Republic of the Philippines
DON HONORIO VENTURA STATE UNIVERSITY
Bacolor, Pampanga

COLLEGE OF COMPUTING STUDIES

SYSTEM ADMINISTRATION AND MAINTENANCE (SAM413)

MODULE 4

Administrative Tasks and Simple Shell Script

(Week 9-10)

Module 4: Administrative Tasks and Simple Shell Script

Foreword

In this module, AWS benefits, features, importance, and advantages are discussed. The step by step how to set up an AWS EC2 instance. The consideration and need to understand what tier to choose in creating an instance. The feature of AWS are discussed according what is their role and importance. Basic XML script to configure the IIS and specific website.

Objectives

At the end of the lesson, students should be able to:

1. recognize, review and discuss the AWS benefits, features, importance, and advantages;
2. recognize and differentiate the different tier to choose in creating instance;
3. recognize, review and discuss the step by step basic XML scripting to configure of IIS and websites;
4. recognize, review and discuss the step by step creating of web server (instance) in AWS EC2;
5. share ideas on the topic under module 4.

Introduction

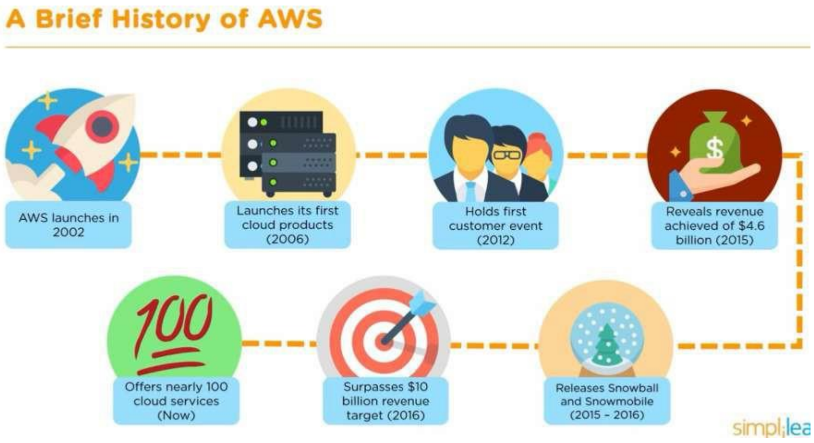
What is Amazon EC2 Instance?

An EC2 instance is nothing but a virtual server in Amazon Web services terminology. It stands for Elastic Compute Cloud. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud. AWS means Amazon Web Services that is used by millions, and to get the answer to this question, we must know that AWS is a cloud provider. It is a safe cloud services platform that offers almost all that a business requires to develop sophisticated applications with reliability, scalability, and flexibility. It is a model for billing generally referred to as “pay-as-you-go,” having no upfront or capital cost. Amazon offers almost 100 services based on-demand, and the list has been rising daily. Operation is almost immediately, and it’s accessible with reduced setup. To master AWS is not all about the online building of sites. The service affords developers access to an interconnected set of attributes offering calculated database storage, power, content delivery, and an increasing portfolio of connected functionality. Organizations around the globe use AWS to develop and scale. Cloud computing has come to remain, and the available solutions from AWS are fast-tracking its development.

Offers computing power
Database storage
aws
Content Delivery
A secure cloud services platform
WHICH WORKS WITH PAY-AS-YOU-GO PRICING

History of AWS

Amazon Web Services was launched in 2002. The company intended to sell the infrastructure that is not in use as a service or offering it to customers, wherein the purpose was met enthusiastically. Amazon had its first AWS product launched in the year 2006. After four years, in the year 2012, Amazon had a huge occasion to gather customer input concerning AWS. To date, the organization continues to hold similar events, like Reinvent, that lets customers share feedback concerning AWS. In 2015, Amazon publicized that the revenue of AWS has amounted to $7.8 billion. From then and 2016, measures had been launched by AWS aiding customers to migrate their services to AWS. Such actions, including the growing and appreciating features of AWS, made further economic growth. In the year 2016, Amazon’s revenue augmented to $12.2 billion in 2016. Presently, AWS provides customers with 160 products and services. The number is liable to increase, owing to the rate at which Amazon builds upon and fine-tunes AWS.


A Brief History of AWS
AWS launches in 2002
Launches its first cloud products (2006)
Holds first customer event (2012)
Reveals revenue achieved of $4.6 billion (2015)
Releases Snowball and Snowmobile (2015 - 2016)
Surpasses $10 billion revenue target (2016)
Offers nearly 100 cloud services (Now)

Who Uses AWS?

The number of organizations under the umbrella of AWS via IT environments is more like assessing globally successful organizations. Using AWS, Adobe provides multi-terabyte working environments accessible by its customers by integrating the system using AWS Cloud. Adobe focuses on operating and organizing its software rather than putting effort to manage and deploy the infrastructure. Airbnb, an online connected vacation rental marketplace for property owners and explorers to connect, sustains an enormous infrastructure in Amazon Web Services while using almost all accessible services. Autodesk is developing software for the entertainment industries, engineering, and design. With the use of services such as Amazon S3 and Amazon RDS, Autodesk can increase its tools for learning machines rather than spending much time on infrastructure management. America Online (AOL) uses AWS to close data centers, economize, and withdraw almost 14,000 co-located and in-house servers. BMW uses AWS for its novel application for car connection, taking sensor data from BMW 7-series cars to enable drivers to assess map information dynamically. With Amazon Web Services' use to distribute cloud-based services like office imaging products and mobile print, Canon's imaging products division tends to benefit from quicker deployment times, global accessibility, and lower cost. Comcase uses AWS, the largest cable company globally and the United States' leading internet service provider for scalable hybrid infrastructure and flexibility. Though more of the European Space Agency's work uses satellites, few of its data storage and computing infrastructure are developed on AWS. The Guardian newspaper vastly uses AWS services to control an analytic dashboard primarily used by editors to view trending stories in recent times. Amongst the world's leading business for news organizations is the Financial Times and uses Amazon Redshift for performing its analysis. Redshift can perform its analysis as fast as possible; some even thought it was malfunctioning. They were accustomed to the overnight running of queries. The Times, therefore, discovered that results were accurate, although much faster. General Electric (GE) is presently migrating over 9,000 workloads and 300 disparate systems of ERP – to Amazon Web Services by reducing its data center footprint from 34 to four by 2021. The list is endless, for example, McDonald's, Kellogg's, NASA, Howard Medical School, to mention but a few are profiting from AWS.

Why is AWS so Successful?

According to various reports from companies, principal aims for not only choosing AWS somewhat depending on it for vital measures of their IT infrastructure:

- Security and durability: Amazon Web Services encodes various data, presenting end-to-end confidentiality, including storage.
- Experience - Designers can depend on Amazon's reputable procedures. Including their techniques, tools including recommended top practices are made due to years of experience.
- Flexibility - There is excessive elasticity in Amazon Web Services, letting designers choose the OS language and database.
- Ease of Use- Amazon Web Services is easy to use. Designers can quickly organize and host apps, construct a novel, or even migrate existing apps.
- Scalability - Apps can easily be scaled up or down but, to an extent, depends on user desires.

- Cost savings - Organizations pay for the computing power, including usage of resources and storage, using no lasting commitments.

Services that are commonly used and provided by Amazon Web Services

Various services available on AWS cannot be overemphasized; they are security, tools development, migration, custom engagement, storage, and many more. Below are some of the services that are commonly used by Amazon Web Services:

- **Amazon EC2** - EC2 can be resettable due to the user's necessity, as long as a secured computing capacity is available in the cloud. For example, in situations whereby web traffic changes, this facility can increase its atmosphere, behind the scenes, to three occurrences when necessary and then contract to a single resource when weight reduces.
- **Amazon Elastic Beanstalk** - It consists of several programming languages; such service enables scale and organizes web apps. Just upload the required code, and the Elastic Beanstalk would automatically handle the setting from the capacity provision to load balancing and auto-scaling to app health checking.
- **Amazon LightSail** - Easy to present and manage, Amazon Lightsail is essentially a confidential server and contains all required to launch a project without delay on a computer-generated machine, such as tools for transferring data, SSD-based storage, static IP, and DNS management.
- **Amazon Lambda** - It enables organizations for code running with no use for server management and provisioning. It automatically scales little requests for a day to so many for a second. Organizations pay for the time used for computing and no charges while code isn't running.
- **Amazon Web Services Storage Services -** with the quantity of data organizations collect, storing the data seems to be high in demand. AWS, therefore, helps bring numerous solutions listed below:
    - **Amazon S3** – With S3, data can be stored and retrieved from anywhere, including IoT sensors, mobile apps, websites, and others. There's flexibility in data management, security, and durability for internet storage.
    - **Amazon Glacier** – It's a service for cloud storage meant for storing data for future use, including long-term backups, highly secure options, and low-cost Glacier.
    - **Amazon Elastic Block Store (EBS)** - EBS offers block store volumes for cases of EC2. It is a dependable storage volume attached to whichever running instance in a similar accessibility zone.
    - **Amazon Elastic File System (EFS)** - EFS can be used with the Amazon Web Services Cloud resources and services. It's scalable and straightforward; it's flexible storing of files for on-premise resources. Containing an intuitive interface allows users to build and file configuring systems without troubling the app growth and automatic shrinking when files are being added or even removed.

- **Amazon Web Services Databases** - Amazon Web Services offers databases in dual primary flavors:

- **Amazon RDS** - Easing the process of setting up, operating, and scaling a relational database in the cloud, Amazon RDS provides cost-efficient and resizable capacity while automating time-consuming administrative tasks such as database hardware setup, repairing, and backups. The enhanced service is for memory performance and output/input processes. Amazon RDS gives you the freedom to use your relational database of choice including the most popular open source and commercial agents and amazon relational database built for the cloud, Amazon Aurora, which offers the performance and availability of traditional commercial databases and fraction of the cost. RDS enables you to scale across a global footprint of data with enterprise high availability and disaster recovery no matter the size, it automates many previous cumbersome task, automatic failover, backups at point in time are restored, disaster recover, access management, encryption, secure networking, monitoring and performance optimization. All these and more can be enabled with a few clicks or API codes. Even, highly regulated industries can leverage RDS which means a broad range of compliance certifications.
- **Amazon Redshift** is a data warehousing service allowing users to examine data with SQL and other intelligence business instruments. It can be used in running multifaceted queries compared to terabytes of organized data and obtain results in seconds.

The Future of AWS

As business and artificial intelligence, including IoT, evolve and indeed come into existence on their own, the necessity for data storage, cloud computing, and security would evolve to new levels. Additional services can be developed in the cloud, such as financial markets, healthcare, and other industries that will become more reliant on these technologies. Gratefully, AWS is out and remains to develop scalable and easy solutions for deploying and managing web apps in the cloud. It is evident that there's a bright future and that this cloud has a silver lining. Suppose you're ready to be a part of the future of AWS. In that case, there's a certification course from Simplilearn that would prepare you to be an industry-ready, in-demand AWS solutions architect, with the privilege of firsthand experience with the management of AWS. You will study how IT architecture rules are redefined by cloud computing and how to scale and design Amazon Web Services cloud operations with Amazon's recommended best practices.

Services of AWS

Since its existence, AWS has developed into a vital technological cloud computing. Below are some essential services offered by AWS:

Amazon S3 - It is a tool used for backing up the internet and less costly for storage options in the category of object-storage. The central part of this option is that data stored can be retrieved from virtually anywhere they are needed.

AWS Data Transfer Products - As the term suggests, they are collecting data, transferring data products, and migration that aids the collection of data seamlessly. They can also allow the monitoring and analysis of data in actual time.

Amazon EC2 (Elastic Compute Cloud) - It provides a resizable and secured capacity for computing, depending on your requirements. The service, therefore, is designed to enable web-scale cloud computing more reachable.

Amazon SNS (Simple Notification Services) - It is a tool for delivering notification messages to a significant number of subscribers via SMS or email. Alarms can be sent, including service notifications and other messages proposed to call attention to important information.

Amazon KMS (Key Management System) - It is a security instrument using 256-bit encryption for data. Also safeguarding it from cybercrime/attacks and hackers.

Amazon Lambda - It's for code running depending on a particular event and manages reliant resources. You do not need either provisioning servers or operating, and how much is paid depending on the length of time, it takes in executing your code. It's cost-effective, unlike services that their charges are according to hourly rates.

Route 53 - It is a DNS service in the cloud that doesn't need you to keep a separate DNS account. The aim is to provide a cost-effective and reliable method to route users for businesses to internet apps

Advantages of Amazon Web Services

Unilever, a well-known organization in the world of consumer goods, would be used as an illustration. Unilever had an issue: It required a quicker time-to-market and a consistent environment, and it's spread across 190 countries and uses digital marketing for its products. Its prevailing legacy in the local climate proved unworkable, incapable of catering to altering IT demands. It then moved a portion of its business to Amazon Web Services, and ever since, rollouts have been smooth, provisioning apps have become dependable, and even infrastructure provisioning has increased. The company can also do everything in scaling push-button and Amazon Web Service's backups that are secured to ensure that all company's data is accessible continuously and secured. Unilever is currently developing with Amazon Web Services, kudos to characteristics like rapid rollouts deployment, capacity to produce current reports and securing backups.

Why Use AWS?

Below are the distinctive features that make Amazon's AWS a leader in the cloud market worldwide.

Improved Security

AWS creates a technological platform that is secured and long-lasting. To ensure that your data is secured and reliable, Amazon's data services and centers have different operational and physical security layers. Amazon Web Services also carry out audits regularly to make sure of the security of the infrastructure. They're also best practices implemented in security and certification offers on ways to deploy AWS security features. There's the assurance of confidentiality, availability, and integrity of your data while providing end-to-end security and privacy. In terms of cloud security provided by Amazon's AWS, they are earnest about it. Their recent addition to security services is called the Amazon Detective, making data investigations more efficient and even faster.

Cost-Effectiveness

A very distinguished advantage of AWS that is promising is the pay-as-you-go pricing model. It means that you pay only for a particular service you subscribed to, and it's only for that specific time you need it. It is a step ahead towards a smart-driven product for organizational development.

Amazon Web Services pricing is related to means through which you pay for utilities such as electricity and water. Therefore, you can pay only for consumed services and immediately discontinue their use, and termination fees or extra costs will not be included.

Save when you Reserve

The suggestion holds specific good services, including Amazon RDS and Amazon EC2. At this point, the real cost is rightly related to the discount accumulated, i.e., if you choose the pay, the total price up-front, then you'll get the extreme discount and vice-versa.

Pay Less by Using More

For particular AWS services like S3 or data transfer OUT from EC2, the more the usage, the less you pay per Gigabyte (GB). These are volume-based discounts that enable you to profit in the long run.

AWS Free Tier

Upon creating a new account, over 60 AWS services can be accessed and freely offered. Nevertheless, these free offers are subdivided into three offers depending on the form of product an organization intends to use.

Flexibility and Openness

AWS is an agnostic-platform for operating languages and systems. You can choose the development platform or programming model that can be more favorable to your business. Businesses are entitled to a potential environment that allows them to access services and software required by a specific application. Hence, no limitations or strict protocols while subscribing to Amazon cloud services that aid migration and enable the building of new solutions. Furthermore, programming languages can be used, including databases, architectures, and programming languages you are accustomed to. In like manner, there won't be any requirement for your IT personnel to pick up a set of new skills, and the total time to market and productivity will considerably increase.

Scalability and Elasticity

Amazon Web Services allows you to experiment, innovate and iterate speedily via a huge cloud global infrastructure. Therefore, in leveraging scalability, Amazon Web Service can effortlessly manage the increase of workload by assigning the resources as a result of the demand, that also within minutes. New applications can as well be used instead of waiting for months for hardware while avoiding resource provisioning upfront for projects with lifetimes that are short and variable rates consumption.

* *AWS can allocate resources via API calls instead of buying hardware, maintaining, and setting it up to assign resources to apps. Elastic load balancing and auto-scaling can automatically measure up Amazon cloud-based resources in the event of an unexpected rise in demand and scaling*

them down whereby need is reduced. The Amazon Web Services cloud is also proved to be useful for recurrent jobs at consistent intervals, jobs that are difficult to accomplish, and short-term jobs.

What are AWS EC2, and why are Businesses Choosing it?

Amazon Elastic Compute Cloud (Amazon EC2) gives scalable computing capacity in the AWS cloud. Through Amazon, your requirements are eradicated to participate in upfront hardware, so faster apps can be developed and deployed. Whatever applications you run, you definitely would require services, sometimes you might need more while sometimes you may not, whatever your requirements, it would be great to obtain services quickly and inexpensively. Traditionally, obtaining services could be stressful, EC2 therefore makes it easy to obtain a virtual service quickly and inexpensively by choosing the exact type you want and the template you prefer to use. This could be done with a few clicks. EC2 also offers a flexible pricing option by paying for only what you use, when you stop your instances, you stop paying (on-demand pricing). Announced in 2006, EC2 (Elastic Compute Cloud) allows you to use Amazon's computing environment and control your complete computing resources. You can speedily scale capability based on the computing requirements as Amazon EC2 enables you to acquire and boot new server instances in a few minutes. It safeguards against common occurrences of failure and offers developers the equipment to advance mobile apps that are tough. A good example is GE Appliances, which has consistently been increasing with these EC2 cases' aid.

Noticeable Characteristics of Amazon Web Services EC2?

AWS EC2 has different attributes that aid the development of scalable applications and enterprise-class, that are tough to failure. Below are some of them:

Amazon Elastic Block Store (EBS)

EBS offers consistent storage for cases of Amazon EC2. Amazon EBS volumes are committed to the network and even without being dependent on a case's life. They can be attached as a standard block device to running an Amazon EC2 case or used as the boot division of an Amazon EC2 case and are incredibly dependable.

While using Amazon EC2 as a boot division, AWS EC2 instances can stop and restart as needed. In a particular accessibility zone, Amazon EBS volumes are duplicated automatically.

EBS-Optimized Cases

A few selected Amazon EC2 types of cases can be launched as EBS-optimized cases for a reduced hourly fee. You can launch a few chosen Amazon EC2 instance types as EBS-optimized instances for a low hourly price. EBS-enhanced cases make it possible for EC2 cases to utilize the IOPS provisioned on an EBS volume. Depending on the type of illustration used, throughput anywhere between 425 Mbps and 14,000 Mbps can be provided by EBS-optimized instances between Amazon EC2 and Amazon EBS.

Elastic IP Addresses

Static IP addresses which can be used for active cloud computing are known as a flexible IP address. A flexible IP address isn't related to a particular instance, but with your account and till you let go of it, the address can be controlled. Through the remapping of your public IP addresses programmatically to whichever case in your account, flexible IP addresses allow you to mask availability zone failures or issues. Amazon EC2 allows you to work around matters with software or cases whereby you don't rapidly map your flexible IP address to the point of replacement. Therefore, you don't have to delay DNS to spread to all customers or hold on for a data technician to reconfigure or substitute your host.

Amazon Virtual Private Cloud

A virtual network can be defined whereby you can launch AWS cloud resources that can be performed via a section of the AWS cloud separate from the aid of Amazon VPC. Route tables, network gateways, and creating subnets, and choosing your IP address variety can be configured. This is because the virtual networking environment can be controlled entirely via Amazon VPC. A hardware virtual private network (VPN) can also be created, which connects your corporate data center and VPC in using AWS cloud as an expansion of the corporate data center.

Amazon CloudWatch

Beginning with Amazon EC2, Amazon CloudWatch observes Amazon cloud resources and apps. Writes, network traffic and disk reads, and CPU operation are amongst the metrics provided by Amazon CloudWatch for reflectiveness into general patterns of demand, resource utilization, and functioning performance. Alarms can be set, statistics can be gotten, and graphs can be viewed for your metric data. All you've to do is select the Amazon EC2 cases you wish to be monitored to use Amazon CloudWatch. Also, you can provide your app or data for a business metric. Amazon CloudWatch will gather and store data easily accessed using command line equipment and AWS APIs.

Auto Scaling

In agreement with the situations outlined, auto-scaling allows you to measure the capacity of your Amazon EC2 both up and down. There is cost reduction for the duration of lulls or spikes in demand via auto-scaling that scales up automatically or scales down the instances of the Amazon EC2 you are using. For applications undergoing hourly, daily, or weekly usage variability, automatic scaling is very appropriate. Amazon CloudWatch allows automatic scaling and will enable you to pay only for what you use.

Elastic Load Balancing

It automatically assigns incoming app traffic to numerous Amazon EC2 cases. Replying to incoming app traffic, elastic load balance offers the load balancing ability needed to aid you in getting better tolerance for errors in your applications. Elastic load balancing notifies traffic to strong cases until the cases that are not solid are repaired. For mobile applications' routine, elastic load balancing can be allowed within single or more accessibility zones. Request inactivity and count are the functioning metrics that can be captured with Amazon CloudWatch. Furthermore, you don't need to pay for anything other than the fees charged for elastic load balancing. Airbnb uses Elastic Load Balancing that automatically distributes inward traffic among multiple Amazon EC2 cases

Auto Recovery

Automatic retrieval is an Amazon CloudFront EC2 characteristic that regains automatically every supported case. If case system damage is identified, it is a stress-free task enabling the automatic feature by constructing an AWS CloudWatch alarm. An additional perk of this characteristic is that it streamlines its capability to preserve a case running when the case gets improved on the new hardware. Therefore, instant migration is no longer needed.

Improved Networking

The characteristic of the EC2 improves the packet for a second performance, alongside the provision of lower latencies and network jitter. Improved networking exploits a new network virtualization stack holding power to give effective I/O implementation and reduced CPU deployment. All you have to do is to launch an HVM AMI in VPC while installing a well-matched driver.

VM Import/Export

These particular EC2 characteristics ease the import of simulated machine images from the cases of EC2 source and export to a similar source at any agreed time. Also, you can export any case of EC2 import previously imported each time the need arises. Nevertheless, there is a standard usage threshold further that there are extra charges to use the service.

High-Performance Computing (HPC) Clusters

Organizations using the computational amount of work like app sensitive to network effects can be managed to accomplish the needed top performance with the aid of these AWS EC2 characteristics. C5 cases are constructed to achieve network-intensive functions. They can be shared into clusters, thus giving a low-latency network function that is advanced required for tightly-coupled, node-to-node communication.

Optimized CPU Configurations

Benefits can be gotten from cost advantages, elasticity, and flexibility of Amazon EC2, also obtain the type of high computing performance and increased network obtained via custom-constructed infrastructure even with multifaceted computational workloads like applications sensitive to network. Applications can get reduced-latency network effectiveness required for node-to- node, firmly coupled communication by programmatically launching cluster compute cases, cluster GPU, and maximum memory into groups. Maximum memory, cluster compute cases, and cluster GPU have been considered to offer top-performance network ability. Cluster cases are appropriate for customer apps essential in performing network-intensive processes and considerably provide improved throughput.

Benefits of Amazon EC2

After maneuvering over the plethora of characteristics that EC2 offers, here are some add-on profits to note.

Simple to Begin With

Getting on track with EC2 is quick and straightforward. All that is required is to choose pre-configured software on Amazon Machine Images (AMIs) by visiting the Amazon Web Services marketplace and begin immediately with Amazon EC2. Use the EC2 AWS support or via a one-click launch in installing the software.

Web-Scale Computing With Elasticity

Your app can automatically scale to-and-fro because Amazon EC2 is controlled through web service application programming interfaces (APIs). You can commission any number of server instances at the same time to increase or decrease capacity in a few minutes through Amazon EC2, which, in turn, makes it a good AWS service.

Low Costing

The financial aids of Amazon's scale are delivered to customers. The capacity for computing that you've used is what you pay for, and it's at a reduced rate. Though EC2 is cost-free to try, there're also some cases where you can choose to pay. Regardless, you can only pay for what you've selected to pay for, without any unexpected or concealed expenses.

Complete Control

You can interrelate with the cases as you can interact with whichever machine you have origin access for every one of them. So you regulate your AWS cases. Data can be saved on your boot partition while you stop a case and later restart it with the web service APIs. Through instances of rebooting remotely by web service APIs, your instance's console output can be accessed.

Security

Your computing resources are protected, and the networking functionality made robust by Amazon EC2 functioning with Amazon Virtual Private Cloud (Amazon VPC). You choose the IP variety for the VPC through which your compute cases are positioned. You identify the cases that stay private and those visible to the internet. As well, for more separation, your cases can be run on dedicated hardware. They are called dedicated instances and run on provisioned resources through EC2. Standardized industry encrypted IPsec VPN connections can be used to connect resources in your VPC with an existing IT infrastructure. You can regulate access to the network and from your cases via network access control lists (ACLs) and a group of security. In leveraging private subnets, VPN connections, network ACLs, and outbound safety filtering collection, amongst other progressive networking characteristics, VPC can be created and introduced into it if you do not have an automatic VPC.

Flexible Services for Cloud Hosting

There are several software development packages and functioning system instance forms that you can choose via Amazon EC2. You can select a boot partition size appropriate for the functioning system and application you have in accumulation to case storage, memory configuration, and CPU via AWS EC2.

Dependability

Dependability aids in the speed and foreseeable replacement of cases via the environmental flexibility provided by AWS EC2. The service leverages Amazon's data centers and network infrastructure to offer 99.95% obtainability in all Amazon EC2 regions (in agreement with the Amazon EC2 service equal agreement).

Works in Combination with Other AWS Components

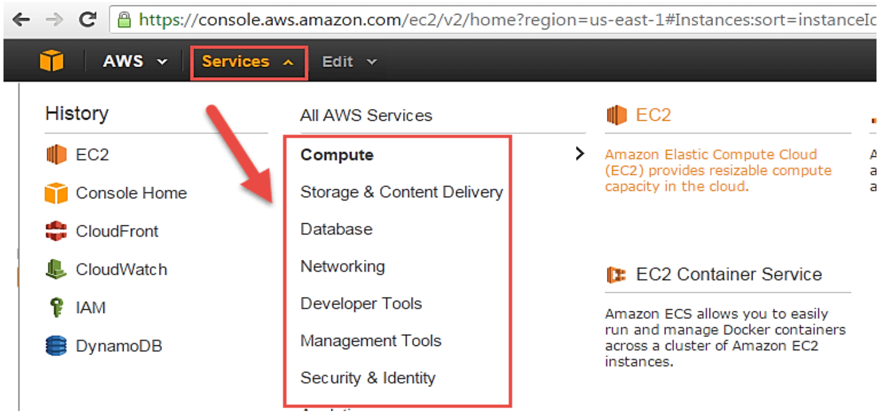
For selecting applications, Amazon EC2 affords computing solutions, query processing and provides storage through working in association with Amazon Simple Queue Service, Amazon Relational Database Service, Amazon Simple DB, and Amazon Simple Storage Service.

How to create an instance in AWS

Login and access to AWS services

Step 1) In this step,

- Login to your AWS account and go to the AWS Services tab at the top left corner.
- Here, you will see all of the AWS Services categorized as per their area viz. Compute, Storage, Database, etc. For creating an EC2 instance, we have to choose Compute EC2 as in the next step.



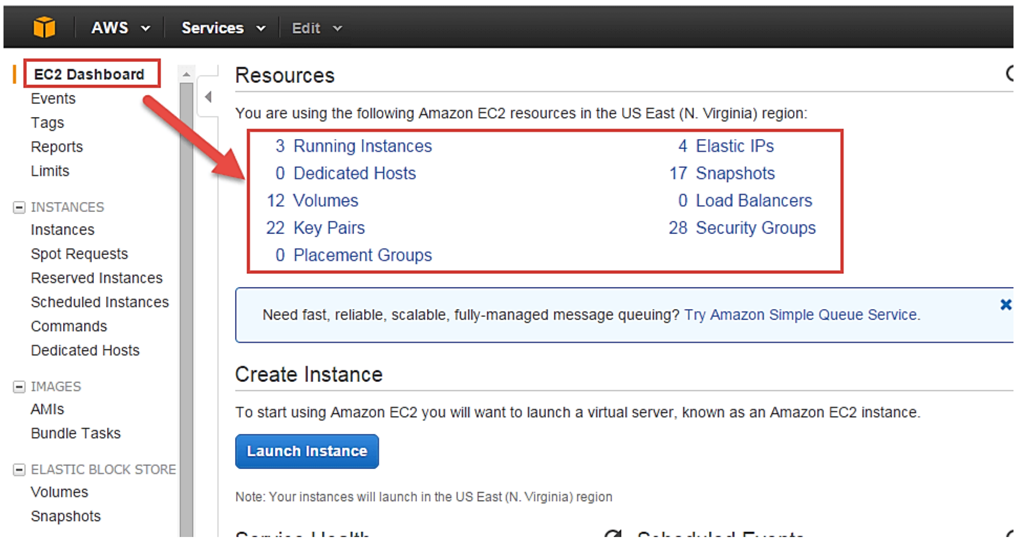
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceIc

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud.

Amazon ECS allows you to easily run and manage Docker containers across a cluster of Amazon EC2 instances.

- Open all the services and click on EC2 under Compute services. This will launch the dashboard of EC2.

Here is the EC2 dashboard. Here you will get all the information in gist about the AWS EC2 resources running.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

3 Running Instances
0 Dedicated Hosts
12 Volumes
22 Key Pairs
0 Placement Groups
4 Elastic IPs
17 Snapshots
0 Load Balancers
28 Security Groups

Need fast, reliable, scalable, fully-managed message queuing? Try Amazon Simple Queue Service.

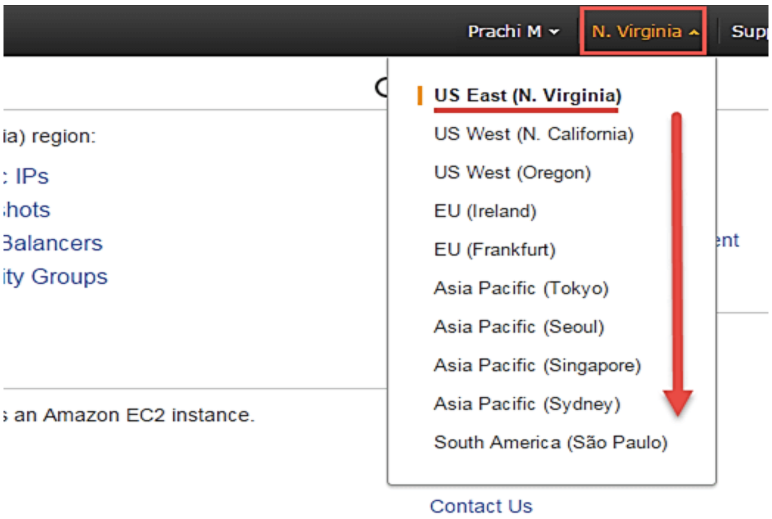
Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US East (N. Virginia) region

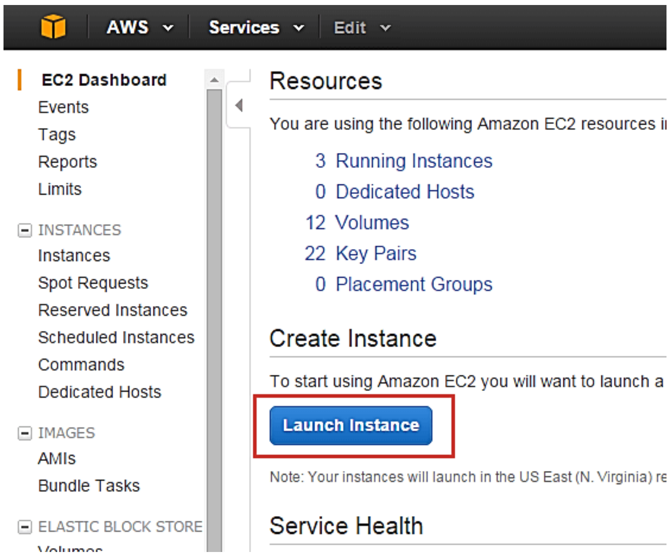
Step 2) On the top right corner of the EC2 dashboard, choose the AWS Region in which you want to provision the EC2 server.

Here we are selecting N. Virginia. AWS provides 10 Regions all over the globe.



Step 3) In this step

- Once your desired Region is selected, come back to the EC2 Dashboard.
- Click on 'Launch Instance' button in the section of Create Instance (as shown below).



Resources

You are using the following Amazon EC2 resources i

3 Running Instances
0 Dedicated Hosts
12 Volumes
22 Key Pairs
0 Placement Groups

Create Instance

To start using Amazon EC2 you will want to launch a

Note: Your instances will launch in the US East (N. Virginia) re

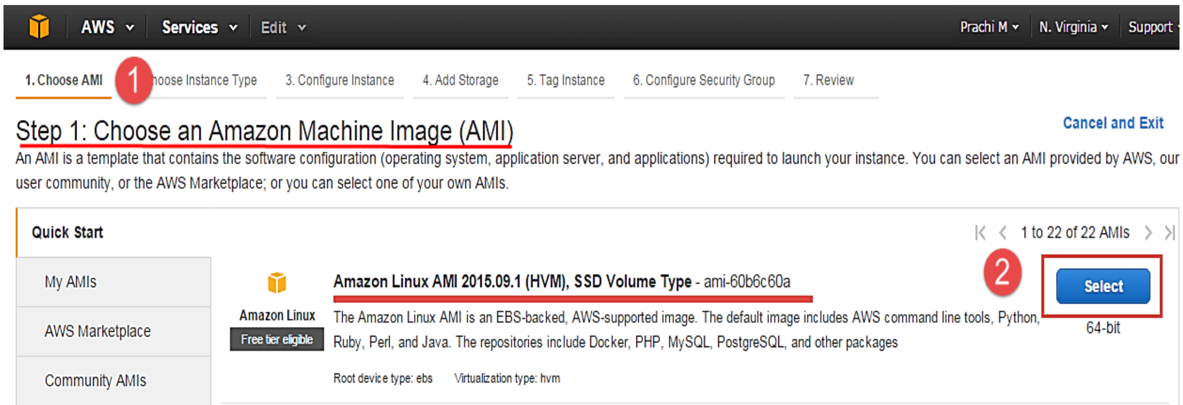
Service Health

- Instance creation wizard page will open as soon as you click ‘Launch Instance’.

Choose AMI

Step 1) In this step we will do,

1. You will be asked to choose an AMI of your choice. (An AMI is an Amazon Machine Image. It is a template basically of an Operating System platform which you can use as a base to create your instance). Once you launch an EC2 instance from your preferred AMI, the instance will automatically be booted with the desired OS. (We will see more about AMIs in the coming part of the tutorial).
2. Here we are choosing the default Amazon Linux (64 bit) AMI.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

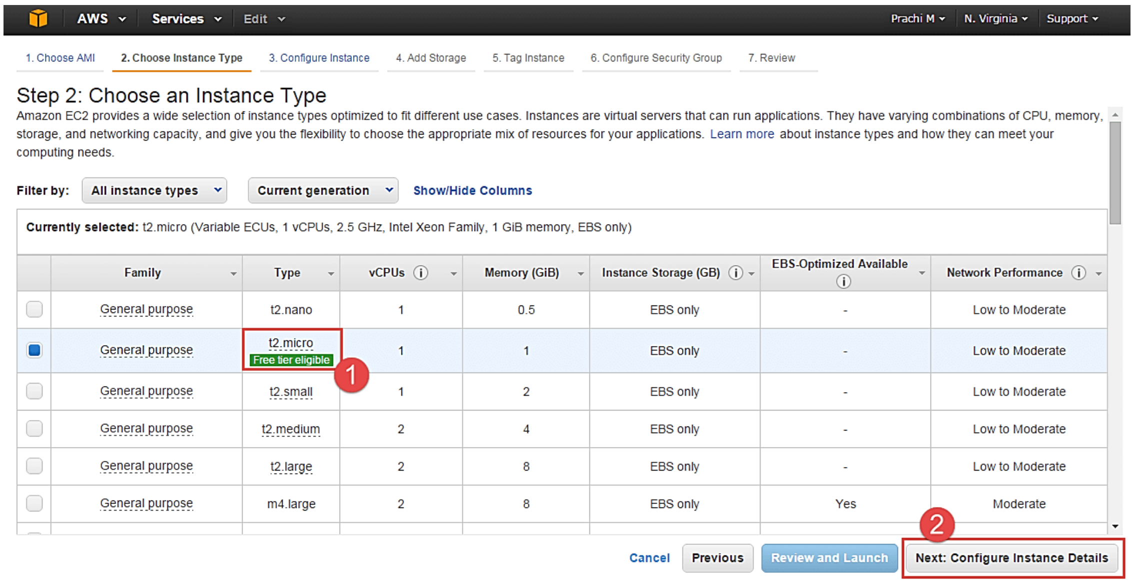
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root device type: ebs Virtualization type: hvm

Choose EC2 Instance Types

Step 1) In the next step, you have to choose the type of instance you require based on your business needs.

1. We will choose t2.micro instance type, which is a 1vCPU and 1GB memory server offered by AWS.
2. Click on “Configure Instance Details” for further configurations



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

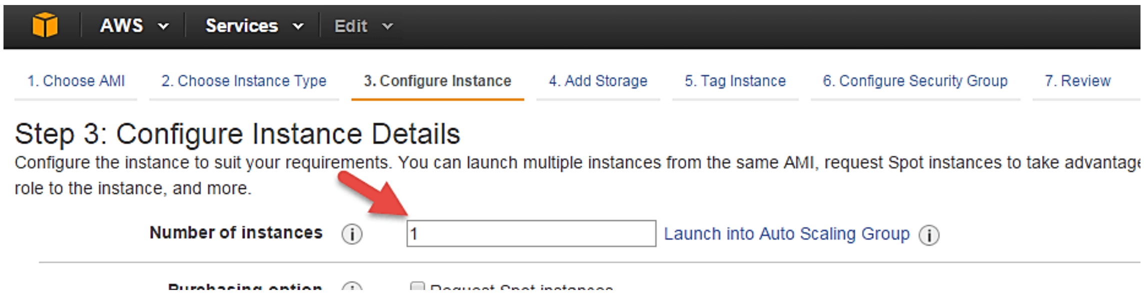
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate
□	General purpose	m4.large	2	8	EBS only	Yes	Moderate

- In the next step of the wizard, enter details like no. of instances you want to launch at a time.
- Here we are launching one instance.

Configure Instance

Step 1) No. of instances- you can provision up to 20 instances at a time. Here we are launching one instance.



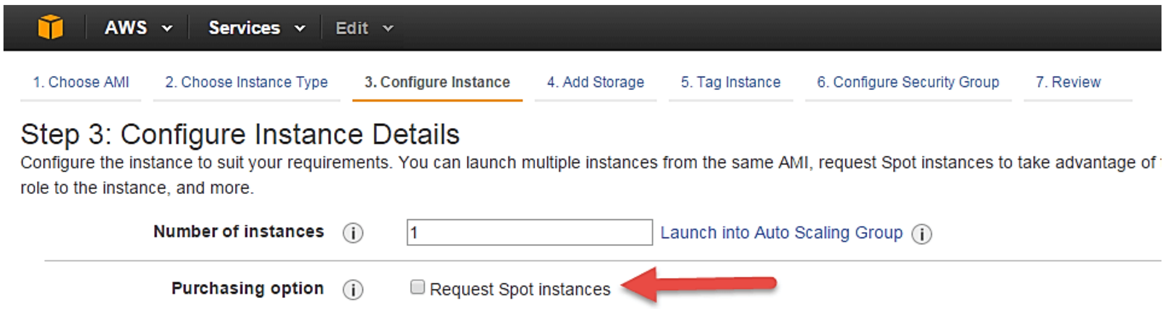
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantag
role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Step 2) Under Purchasing Options, keep the option of 'Request Spot Instances' unchecked as of now. (This is done when we wish to launch Spot instances instead of on-demand ones. We will come back to Spot instances in the later part of the tutorial).



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of
role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

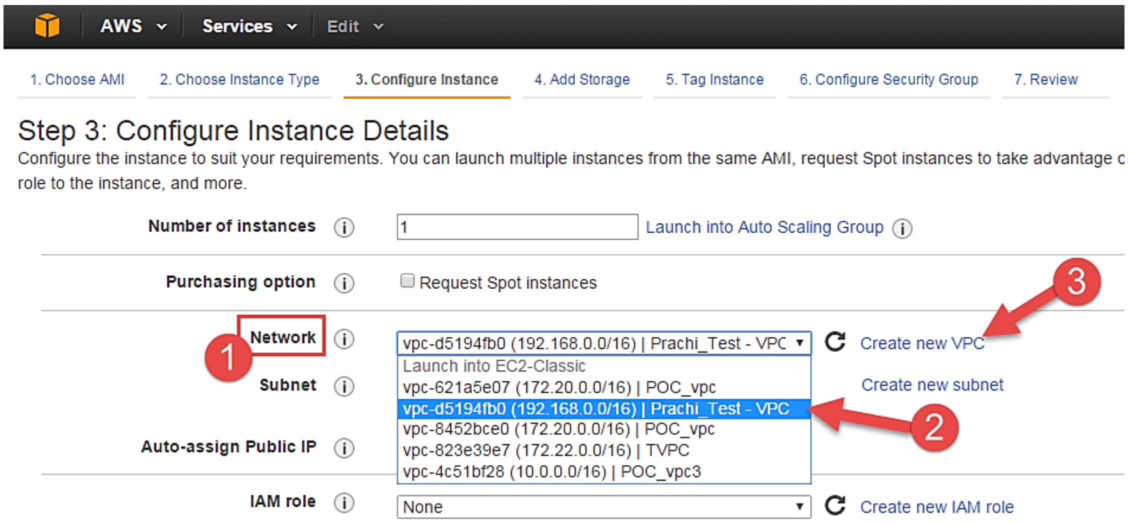
Step 3) Next, we have to configure some basic networking details for our EC2 server.

- You have to decide here, in which VPC (Virtual Private Cloud) you want to launch your instance and under which subnets inside your VPC. It is better to determine and plan this prior to launching the instance. Your AWS architecture set-up should include IP ranges for your subnets etc. pre-planned for better management. (We will see how to create a new VPC in Networking section of the tutorial.
- Subnetting should also be pre-planned. E.g.: If it's a web server you should place it in the public subnet and if it's a DB server, you should place it in a private subnet all inside your VPC.

Below,

1. Network section will give a list of VPCs available in our platform.
2. Select an already existing VPC
3. You can also create a new VPC

Here I have selected an already existing VPC where I want to launch my instance.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage c role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

- Launch into EC2-Classic
- vpc-621a5e07 (172.20.0.0/16) | POC_vpc
- vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC
- vpc-8452bce0 (172.20.0.0/16) | POC_vpc
- vpc-823e39e7 (172.22.0.0/16) | TVPC
- vpc-4c51bf28 (10.0.0.0/16) | POC_vpc3

Subnet

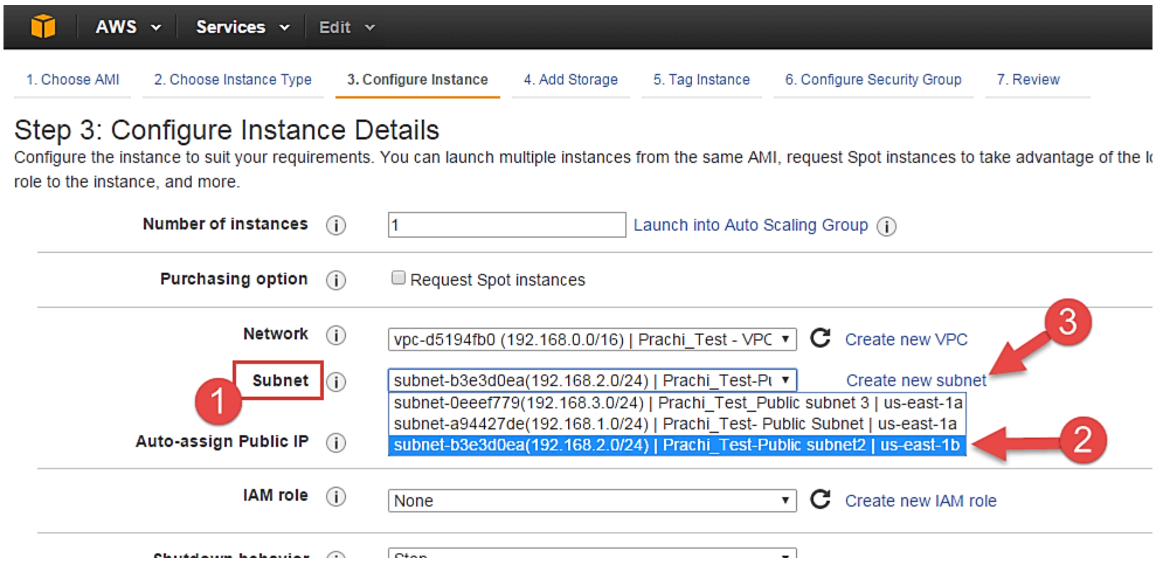
Auto-assign Public IP

IAM role: None

Step 4) In this step,

- A VPC consists of subnets, which are IP ranges that are separated for restricting access.
- Below,

1. Under Subnets, you can choose the subnet where you want to place your instance.
2. I have chosen an already existing public subnet.
3. You can also create a new subnet in this step.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the l role to the instance, and more.

Number of instances: 1

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu

- subnet-0eeef779(192.168.3.0/24) | Prachi_Test_Public subnet 3 | us-east-1a
- subnet-a94427de(192.168.1.0/24) | Prachi_Test- Public Subnet | us-east-1a
- subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Public subnet2 | us-east-1b

Auto-assign Public IP

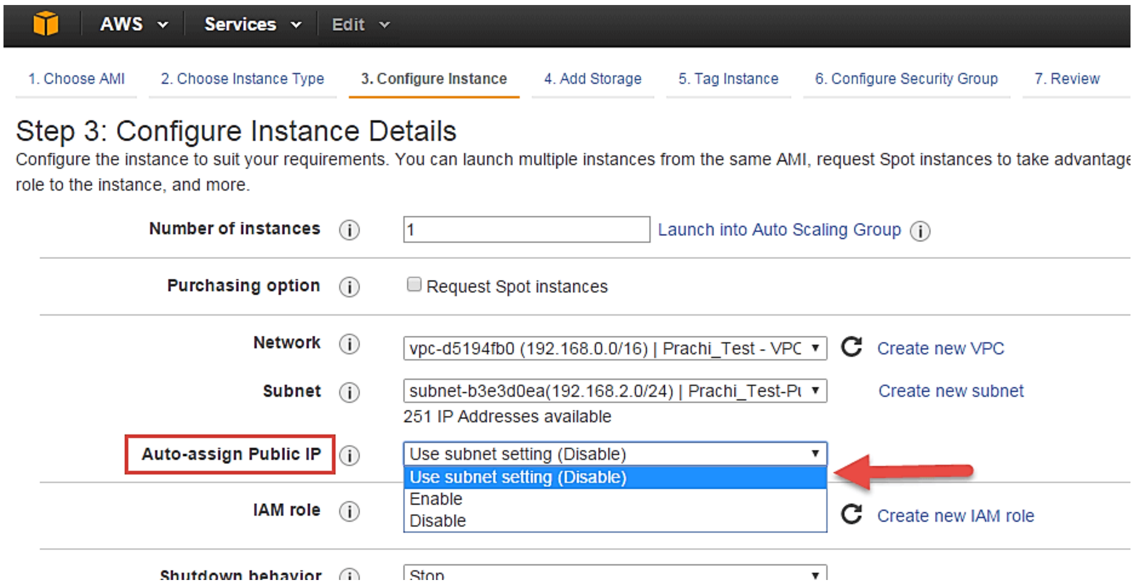
IAM role: None

Shutdown behavior: Stop

- Once your instance is launched in a public subnet, AWS will assign a dynamic public IP to it from their pool of IPs.

Step 5) In this step,

- You can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.
- Here we are going to assign this instance a static IP called as EIP (Elastic IP) later. So we keep this feature disabled as of now.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage
role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet

251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

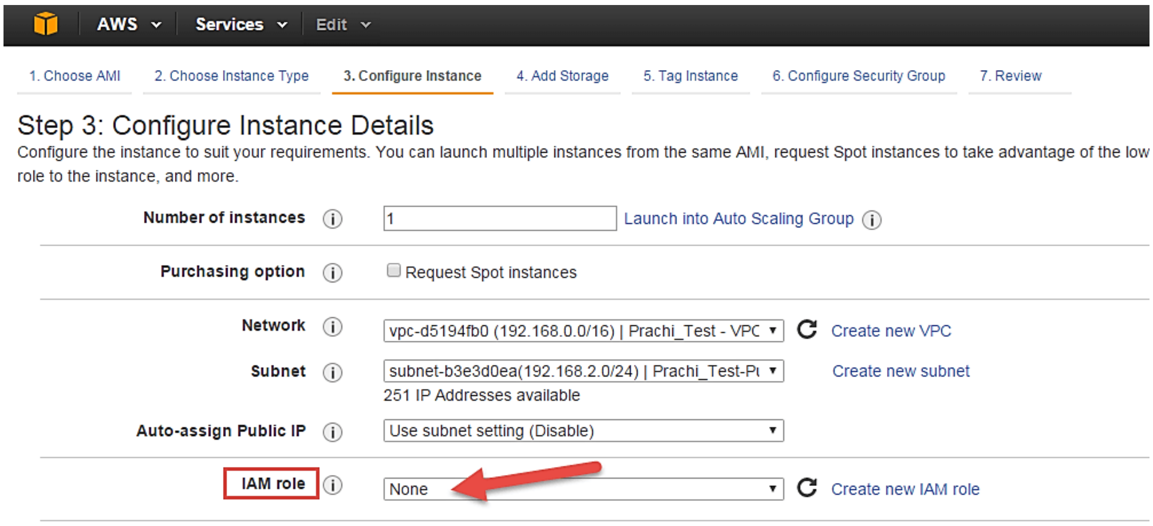
- Use subnet setting (Disable)
- Enable
- Disable

IAM role — Create new IAM role

Shutdown behavior: Stop

Step 6) In this step,

- In the following step, keep the option of IAM role ‘None’ as of now. We will visit the topic of IAM role in detail in IAM services.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the low
role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pu — Create new subnet

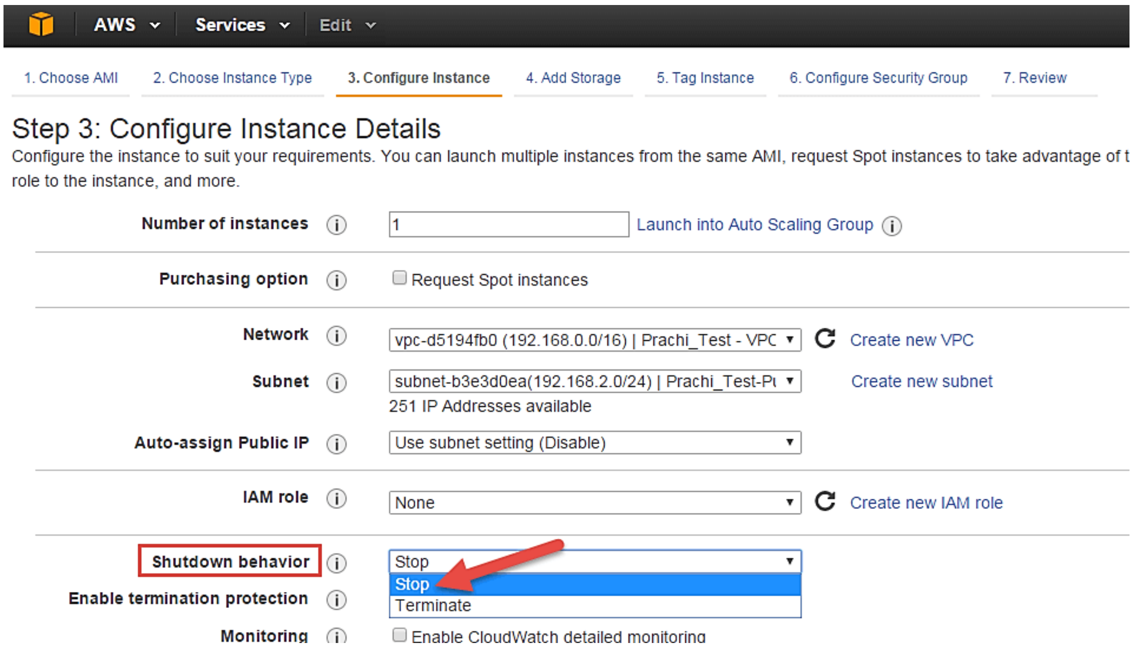
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None — Create new IAM role

Step 7) In this step, you have to do following things

- Shutdown Behavior – when you accidently shut down your instance, you surely don’t want it to be deleted but stopped.
- Here we are defining my shutdown behavior as Stop.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of t role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pι — Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None — Create new IAM role

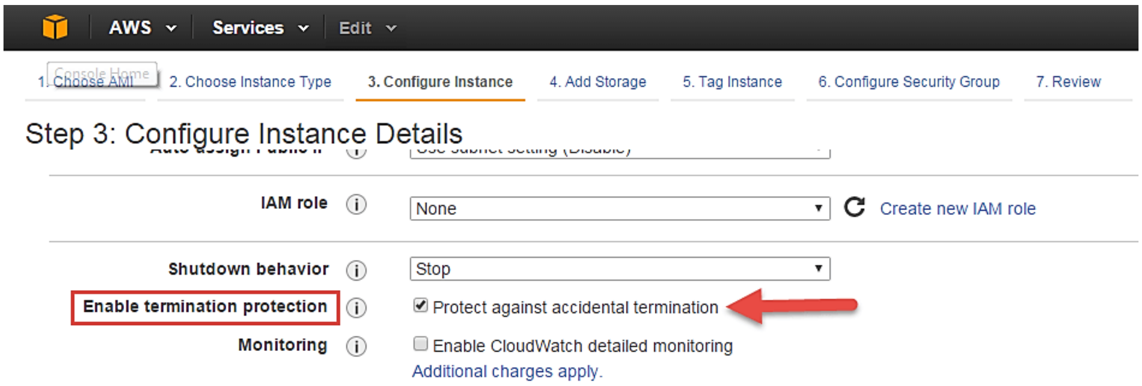
Shutdown behavior: Stop
Stop
Terminate

Enable termination protection

Monitoring: □ Enable CloudWatch detailed monitoring

Step 8) In this step,

- In case, you have accidently terminated your instance, AWS has a layer of security mechanism. It will not delete your instance if you have enabled accidental termination protection.
- Here we are checking the option for further protecting our instance from accidental termination.



Step 3: Configure Instance Details

IAM role: None — Create new IAM role

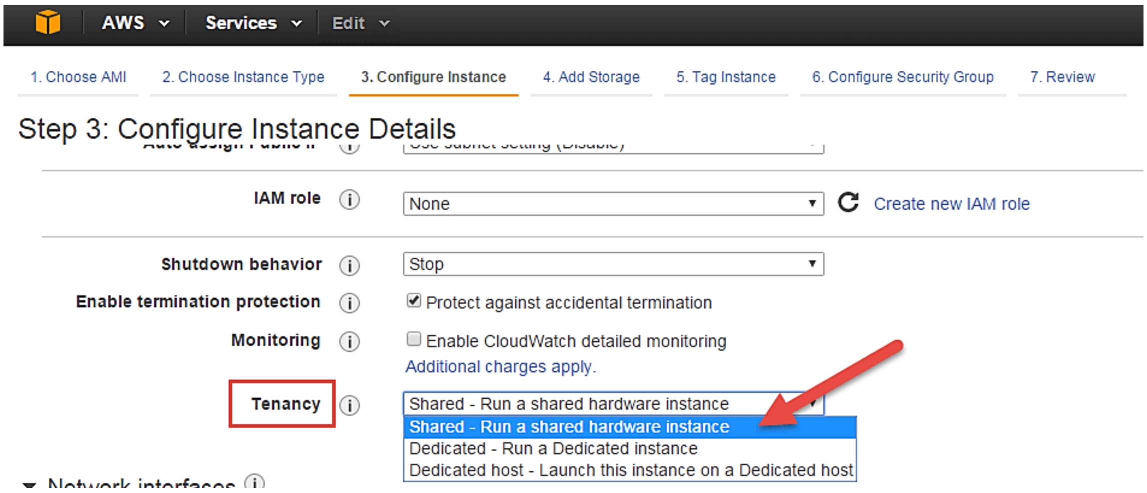
Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Step 9) In this step,

- Under Monitoring- you can enable Detailed Monitoring if your instance is a business critical instance. Here we have kept the option unchecked. AWS will always provide Basic monitoring on your instance free of cost. We will visit the topic of monitoring in AWS Cloud Watch part of the tutorial.
- Under Tenancy- select the option if shared tenancy. If your application is a highly secure application, then you should go for dedicated capacity. AWS provides both options.



Step 3: Configure Instance Details

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

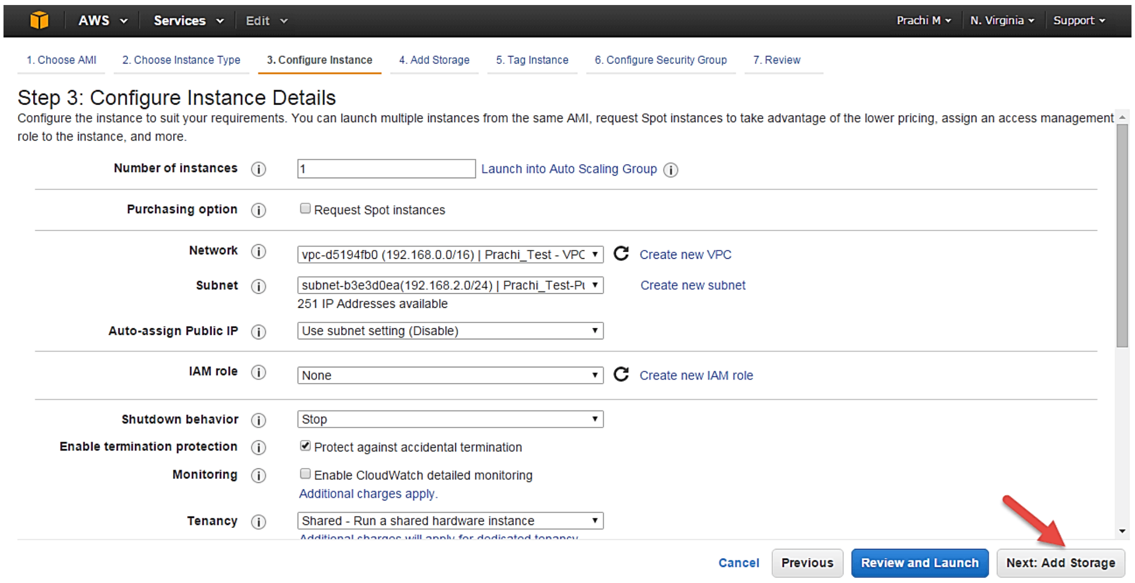
Monitoring: ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance
- Shared - Run a shared hardware instance
- Dedicated - Run a Dedicated instance
- Dedicated host - Launch this instance on a Dedicated host

Network interfaces

Step 10) In this step,

- Click on 'Add Storage' to add data volumes to your instance in next step.



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-d5194fb0 (192.168.0.0/16) | Prachi_Test - VPC — Create new VPC

Subnet: subnet-b3e3d0ea(192.168.2.0/24) | Prachi_Test-Pı — Create new subnet
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Disable)

IAM role: None — Create new IAM role

Shutdown behavior: Stop

Enable termination protection: ☑ Protect against accidental termination

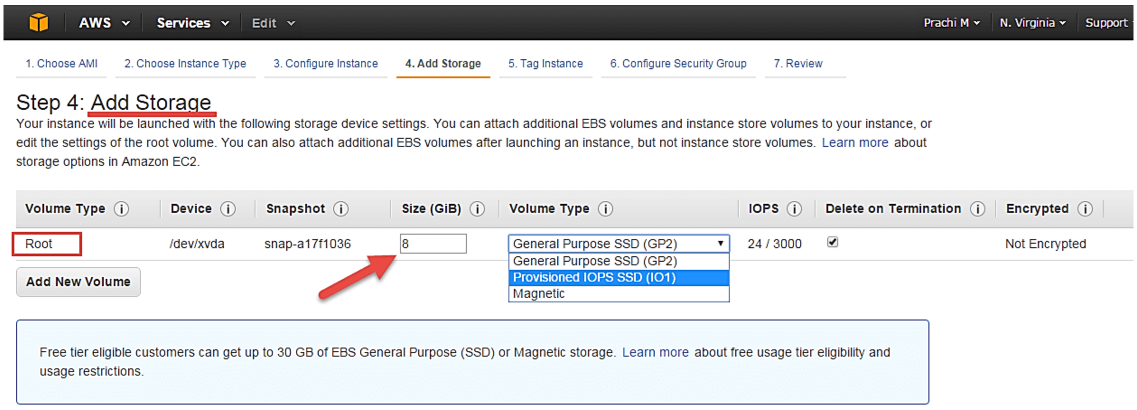
Monitoring: ☐ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared - Run a shared hardware instance

Add Storage

Step 1) In this step we do following things,

- In the Add Storage step, you'll see that the instance has been automatically provisioned a General Purpose SSD root volume of 8GB. (Maximum volume size we can give to a General Purpose volume is 16GB)
- You can change your volume size, add new volumes, change the volume type, etc.
- AWS provides 3 types of EBS volumes- Magnetic, General Purpose SSD, Provisioned IOPs. You can choose a volume type based on your application's IOPs needs.



Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/xvda	snap-a17f1036	8	General Purpose SSD (GP2) General Purpose SSD (GP2) Provisioned IOPS SSD (IO1) Magnetic	24 / 3000	☑	Not Encrypted

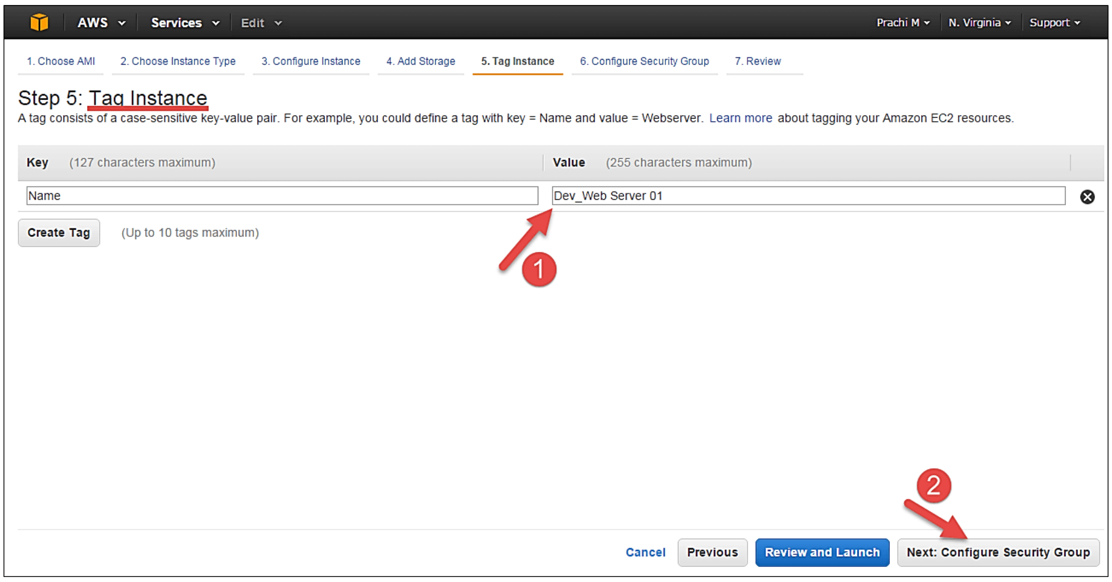
Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

Tag Instance

Step 1) In this step

- you can tag your instance with a key-value pair. This gives visibility to the AWS account administrator when there are lot number of instances.
- The instances should be tagged based on their department, environment like Dev/SIT/Prod. Etc. this gives a clear view of the costing on the instances under one common tag.

1. Here we have tagged the instance as a **Dev_Web server 01**
2. Go to configure Security Groups later



Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	Dev_Web Server 01

(Up to 10 tags maximum)

Configure Security Groups

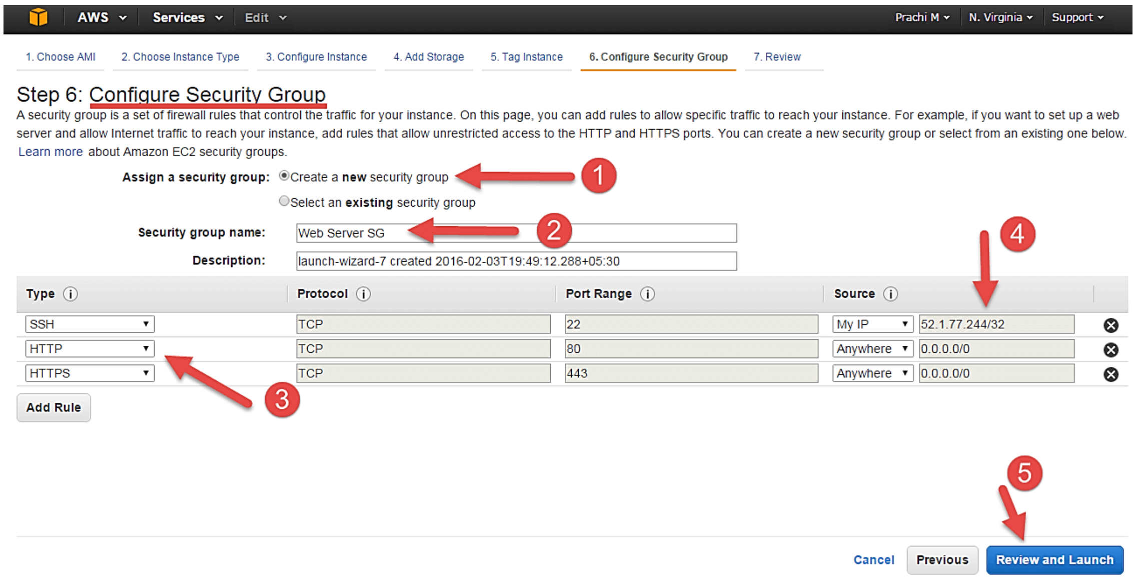
Step 1) In this next step of configuring Security Groups, you can restrict traffic on your instance ports. This is an added firewall mechanism provided by AWS apart from your instance's OS firewall.

You can define open ports and IPs.

- Since our server is a webserver=, we will do following things

1. Creating a new Security Group
2. Naming our SG for easier reference
3. Defining protocols which we want enabled on my instance

4. Assigning IPs which are allowed to access our instance on the said protocols
5. Once, the firewall rules are set- Review and launch



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

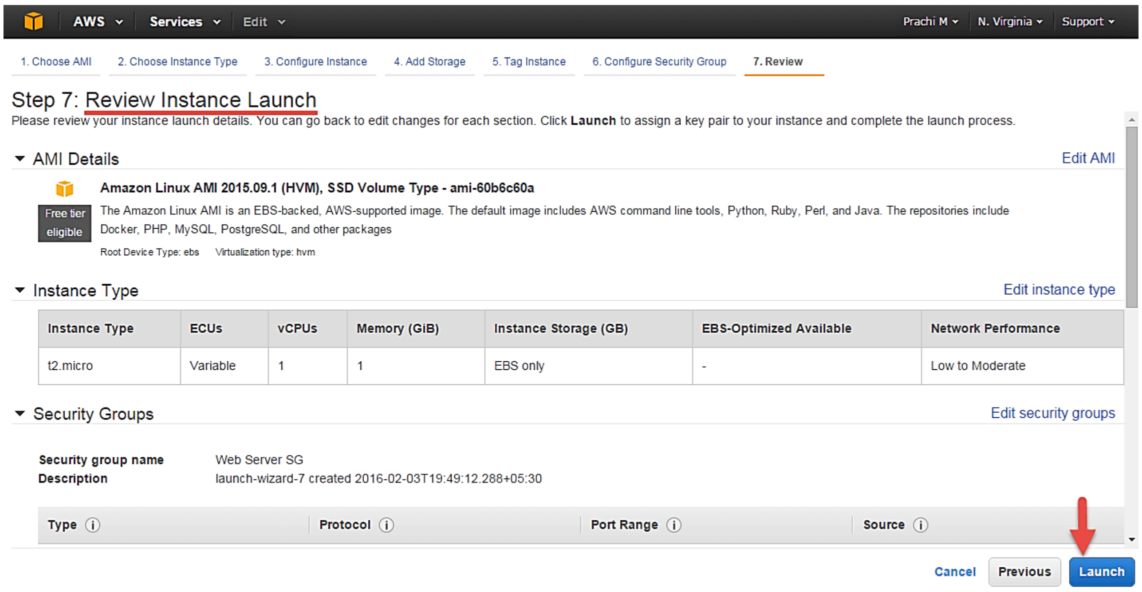
Security group name: Web Server SG

Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source	
SSH	TCP	22	My IP	52.1.77.244/32
HTTP	TCP	80	Anywhere	0.0.0.0/0
HTTPS	TCP	443	Anywhere	0.0.0.0/0

Review Instances

Step 1) In this step, we will review all our choices and parameters and go ahead to launch our instance.



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type - ami-60b6c60a

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

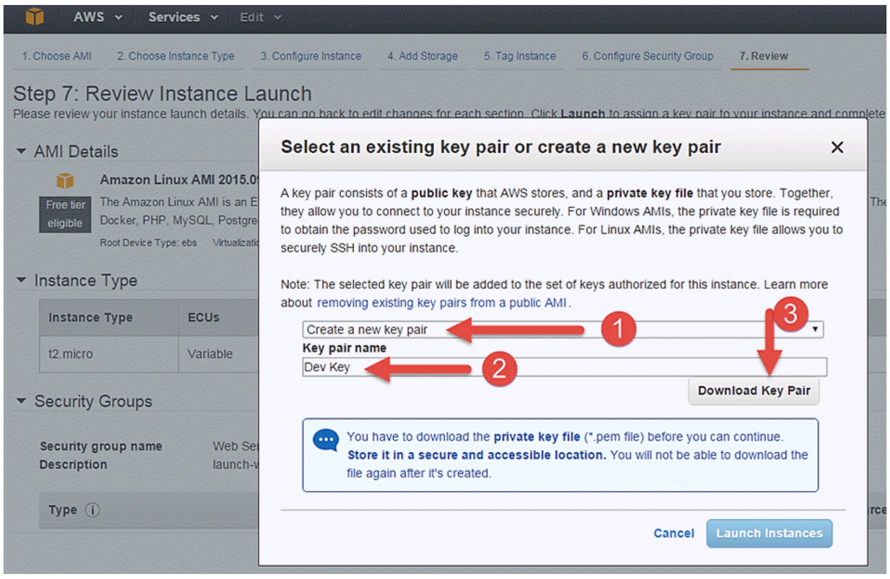
Security group name: Web Server SG
Description: launch-wizard-7 created 2016-02-03T19:49:12.288+05:30

Type	Protocol	Port Range	Source

Step 2) In the next step you will be asked to create a key pair to login to you an instance. A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the private key. Make sure you download the key and keep it safe and secured; if it is lost you cannot download it again.

1. Create a new key pair
2. Give a name to your key
3. Download and save it in your secured folder



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair ▾

Key pair name

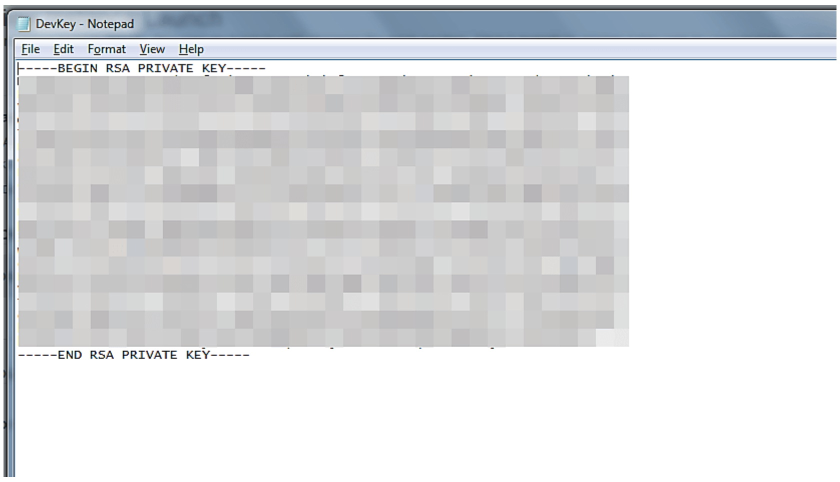
Dev Key

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

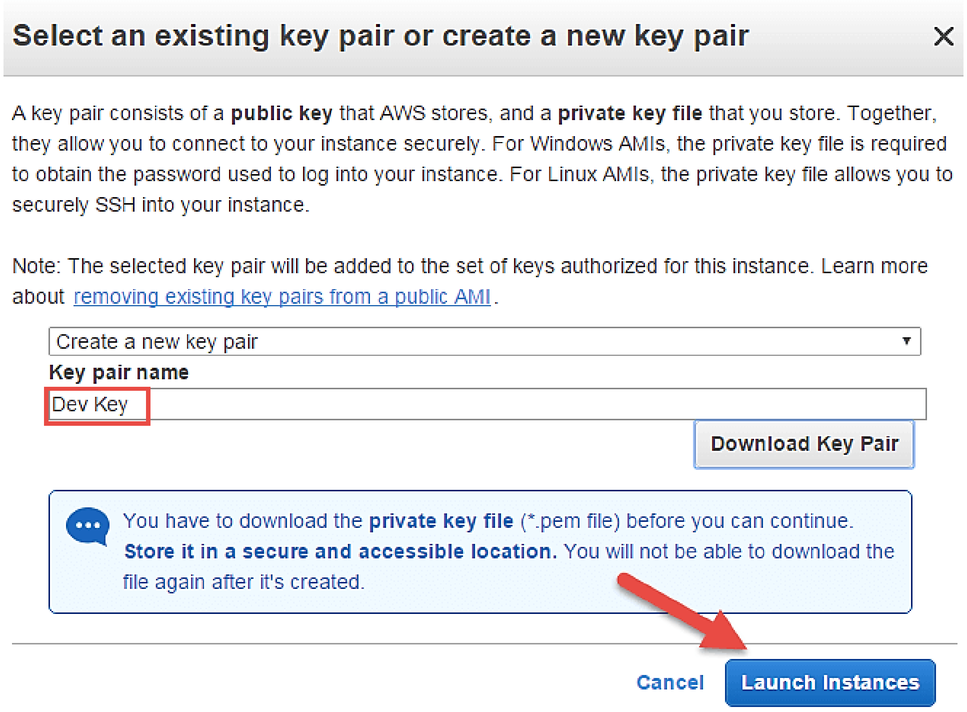
Cancel Launch Instances

- When you download your key, you can open and have a look at your RSA private key.



```
-----BEGIN RSA PRIVATE KEY-----
[redacted]
-----END RSA PRIVATE KEY-----
```

Step 3) Once you are done downloading and saving your key, launch your instance.



Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Create a new key pair ▾

Key pair name

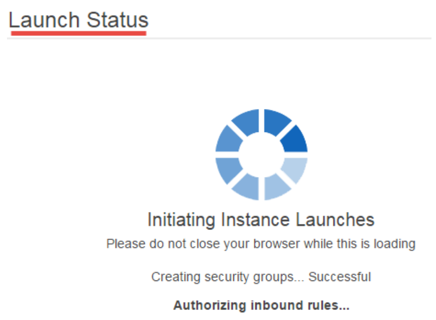
Dev Key

Download Key Pair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel Launch Instances

- You can see the launch status meanwhile.



Launch Status

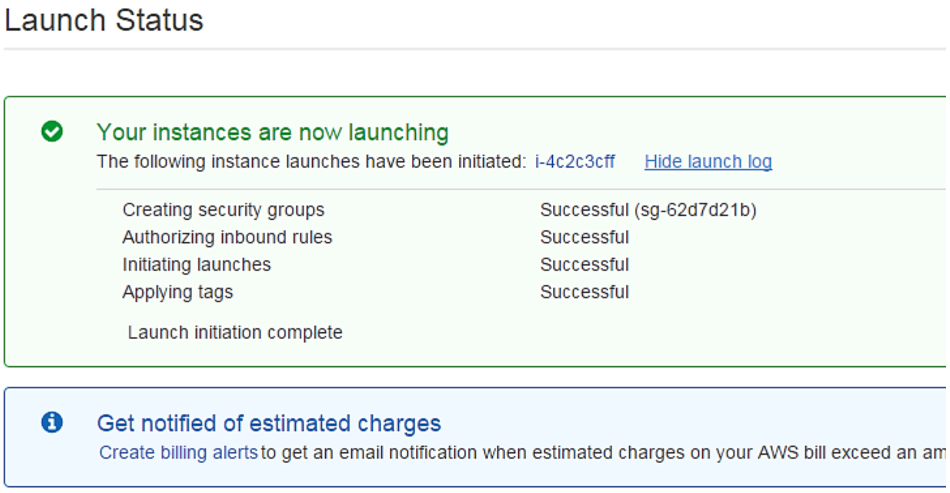
Initiating Instance Launches

Please do not close your browser while this is loading

Creating security groups... Successful

Authorizing inbound rules...

- You can also see the launch log.



Launch Status

Your instances are now launching

The following instance launches have been initiated: i-4c2c3cff Hide launch log

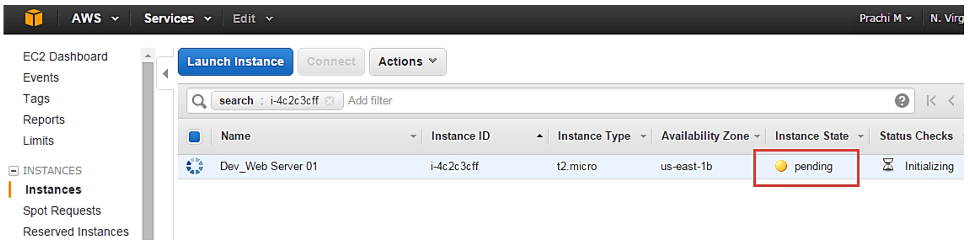
Creating security groups	Successful (sg-62d7d21b)
Authorizing inbound rules	Successful
Initiating launches	Successful
Applying tags	Successful

Launch initiation complete

Get notified of estimated charges

Create billing alerts to get an email notification when estimated charges on your AWS bill exceed an am

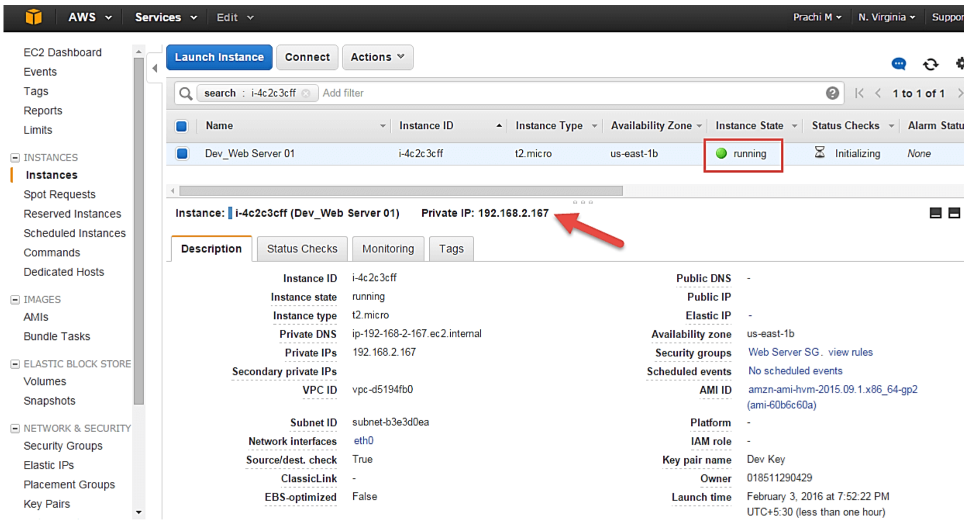
- Click on the ‘Instances’ option on the left pane where you can see the status of the instance as ‘Pending’ for a brief while.



search : i-4c2c3cff Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	pending	Initializing

- Once your instance is up and running, you can see its status as ‘Running’ now.
- Note that the instance has received a Private IP from the pool of AWS.



search : i-4c2c3cff Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Statu
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	running	Initializing	None

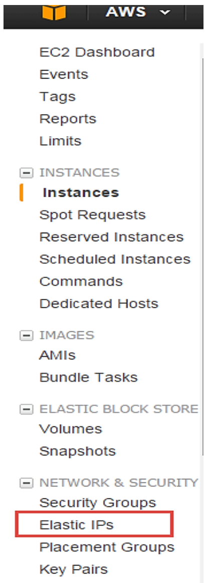
Instance: i-4c2c3cff (Dev_Web Server 01) Private IP: 192.168.2.167

Instance ID	i-4c2c3cff	Public DNS	-
Instance state	running	Public IP	
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-192-168-2-167.ec2.internal	Availability zone	us-east-1b
Private IPs	192.168.2.167	Security groups	Web Server SG. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-d5194fb0	AMI ID	amzn-ami-hvm-2015.09.1.x86_64-gp2 (ami-60b6c60a)
Subnet ID	subnet-b3e3d0ea	Platform	-
Network interfaces	eth0	IAM role	-
Source/dest. check	True	Key pair name	Dev Key
ClassicLink	-	Owner	018511290429
EBS-optimized	False	Launch time	February 3, 2016 at 7:52:22 PM UTC+5:30 (less than one hour)

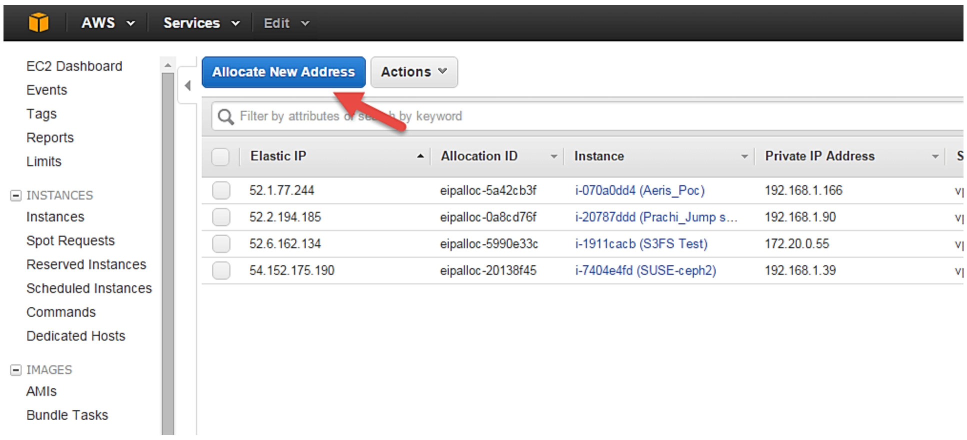
Create a EIP and connect to your instance

An EIP is a static public IP provided by AWS. It stands for Elastic IP. Normally when you create an instance, it will receive a public IP from the AWS's pool automatically. If you stop/reboot your instance, this public IP will change- it'dynamic. In order for your application to have a static IP from where you can connect via public networks, you can use an EIP.

Step 1) On the left pane of EC2 Dashboard, you can go to 'Elastic IPs' as shown below.

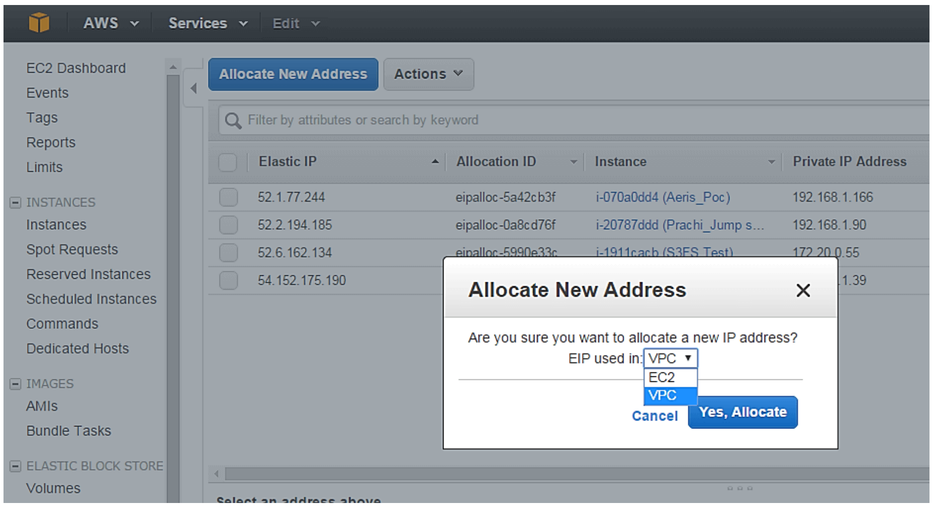


Step 2) Allocate a new Elastic IP Address.



Elastic IP	Allocation ID	Instance	Private IP Address	S
52.1.77.244	eipalloc-5a42cb3f	i-070a0dd4 (Aeris_Poc)	192.168.1.166	v
52.2.194.185	eipalloc-0a8cd76f	i-20787ddd (Prachi_Jump s…	192.168.1.90	v
52.6.162.134	eipalloc-5990e33c	i-1911cacb (S3FS Test)	172.20.0.55	v
54.152.175.190	eipalloc-20138f45	i-7404e4fd (SUSE-ceph2)	192.168.1.39	v

Step 3) Allocate this IP to be used in a VPC scope.



Elastic IP	Allocation ID	Instance	Private IP Address
52.1.77.244	eipalloc-5a42cb3f	i-070a0dd4 (Aeris_Poc)	192.168.1.166
52.2.194.185	eipalloc-0a8cd76f	i-20787ddd (Prachi_Jump s…	192.168.1.90
52.6.162.134	eipalloc-5990e33c	i-1911cacb (S3FS Test)	172.20.0.55
54.152.175.190			1.39

Allocate New Address

Are you sure you want to allocate a new IP address?

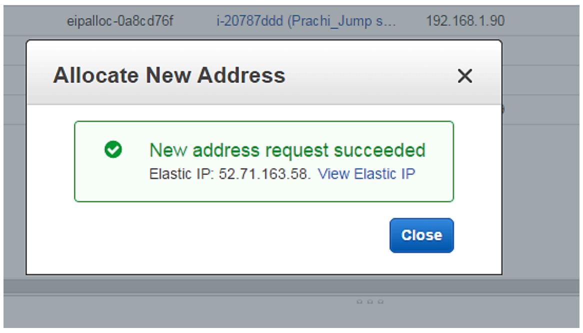
EIP used in: VPC

EC2
VPC

Cancel Yes, Allocate

Select an address above

- Your request will succeed if you don’t have 5 or more than 5 EIPs already in your account.



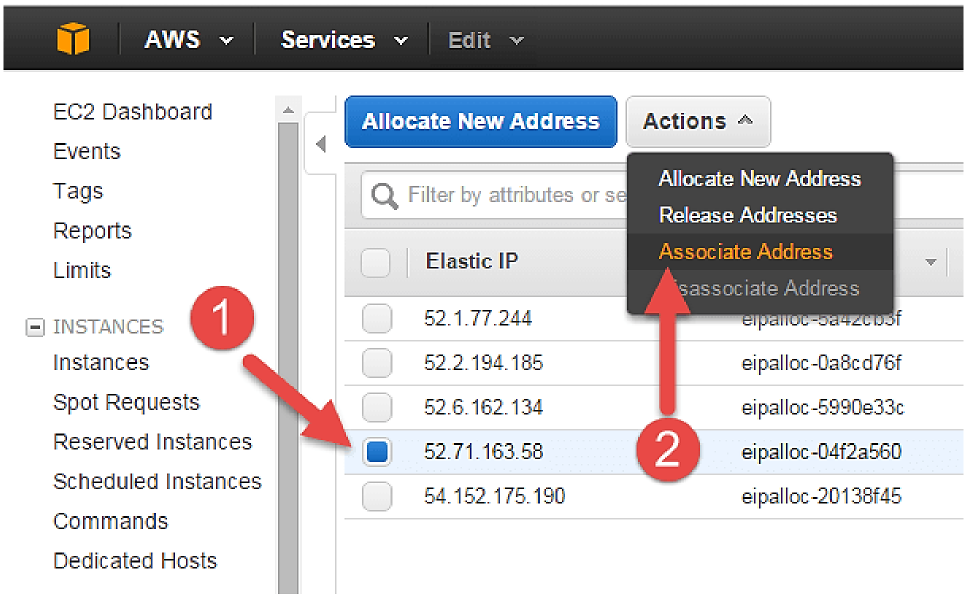
eipalloc-0a8cd76f i-20787ddd (Prachi_Jump s... 192.168.1.90

Allocate New Address

New address request succeeded
Elastic IP: 52.71.163.58. View Elastic IP

Step 4) Now assign this IP to your instance.

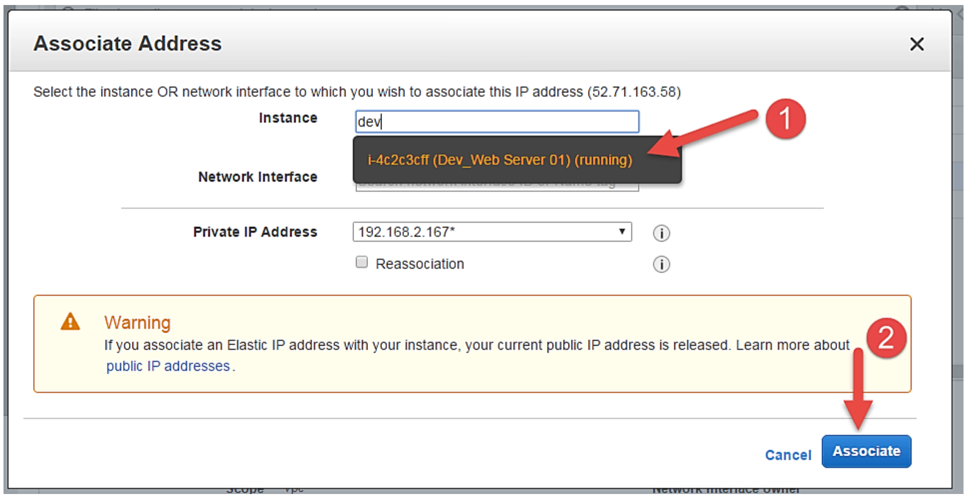
1. Select the said IP
2. Click on Actions **->** Associate Address



Elastic IP	
52.1.77.244	eipalloc-5a42cb3f
52.2.194.185	eipalloc-0a8cd76f
52.6.162.134	eipalloc-5990e33c
52.71.163.58	eipalloc-04f2a560
54.152.175.190	eipalloc-20138f45

Step 5) In the next page,

1. Search for your instance and
2. Associate the IP to it.



Associate Address

Select the instance OR network interface to which you wish to associate this IP address (52.71.163.58)

Instance: dev
i-4c2c3cff (Dev_Web Server 01) (running)

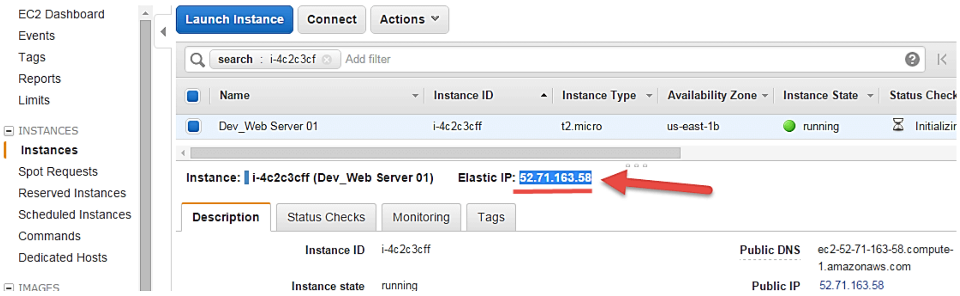
Network Interface

Private IP Address: 192.168.2.167*

□ Reassociation

Warning
If you associate an Elastic IP address with your instance, your current public IP address is released. Learn more about public IP addresses.

Step 6) Come back to your instances screen, you'll see that your instance has received your EIP.



search : i-4c2c3cf

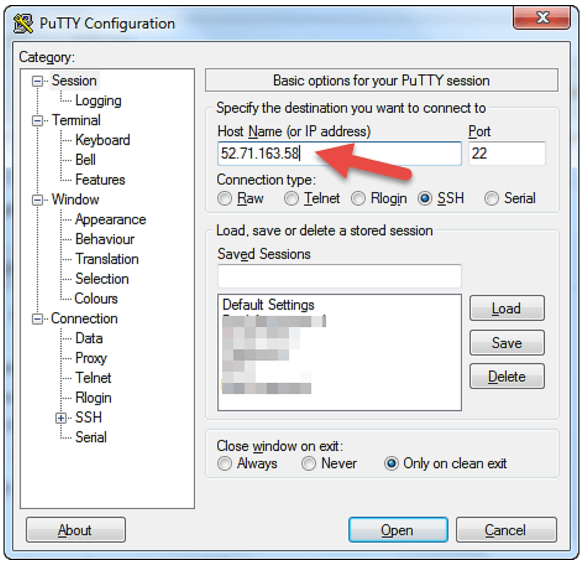
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Check
Dev_Web Server 01	i-4c2c3cff	t2.micro	us-east-1b	running	Initializir

Instance: i-4c2c3cff (Dev_Web Server 01) Elastic IP: 52.71.163.58

Instance ID i-4c2c3cff Public DNS ec2-52-71-163-58.compute-1.amazonaws.com

Instance state running Public IP 52.71.163.58

Step 7) Now open putty from your programs list and add your same EIP in there as below.



Host Name (or IP address): 52.71.163.58 Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

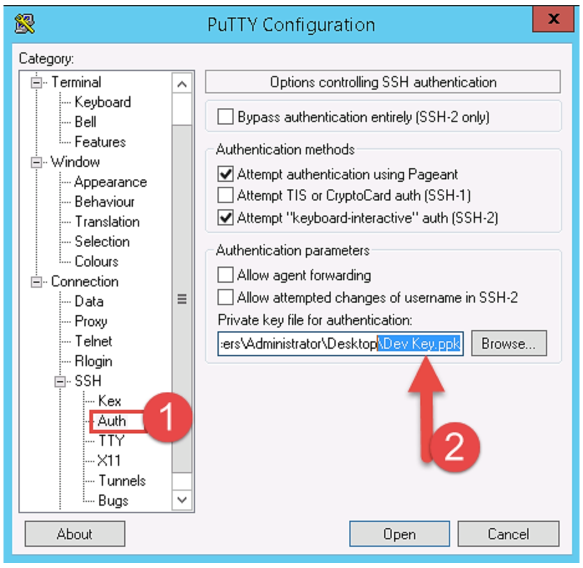
Close window on exit: ○ Always ○ Never ● Only on clean exit

Step 8) In this step,

Add your private key in putty for secure connection

1. Go to Auth
2. Add your private key in .ppk (putty private key) format. You will need to convert pem file from AWS to ppk using puttygen

Once done click on "Open" button



☐ Bypass authentication entirely (SSH-2 only)

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

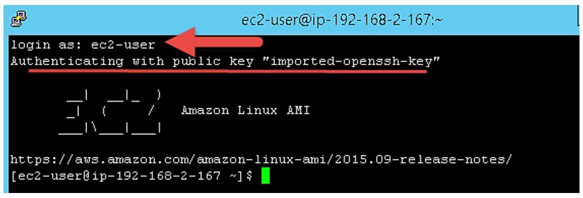
☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication: :ers\Administrator\Desktop\Dev Key.ppk

- Once you connect, you will successfully see the Linux prompt.

- Please note that the machine you are connecting from should be enabled on the instance Security Group for SSH (like in the steps above).



```
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.09-release-notes/
[ec2-user@ip-192-168-2-167 ~]$
```

Once you become familiar with the above steps for launching the instance, it becomes a matter of 2 minutes to launch the same!

You can now use your on-demand EC2 server for your applications.

What is Spot Instance?

A spot Instance is an offering from AWS; it allows an AWS business subscriber to bid on unused AWS compute capacity. The hourly price for a Spot instance is decided by AWS, and it fluctuates depending on the supply and demand for Spot instances.

Your Spot instance runs whenever your bid exceeds the current market price. The price of a spot instance varies based on the instance type and the Availability Zone in which the instance can be provisioned.

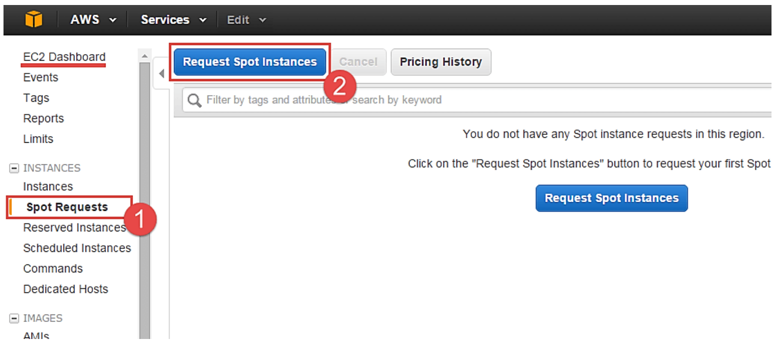
When your bid price exceeds the market spot price of the instance called as the 'spot price,' your instance stays running. When the spot price overshoots the bid price, AWS will terminate your instance automatically. Therefore, it is necessary to plan the spot instances in your application architecture carefully.

Create a Spot Request

In order to launch a spot instance, you have to first create a Spot Request.

Follow the steps below to create a Spot Request.

1. On the EC2 Dashboard select 'Spot Requests' from the left pane under Instances.
2. Click on the button 'Request Spot Instances" as shown below.



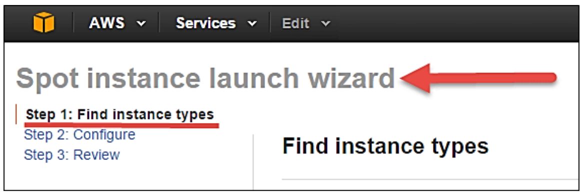
You do not have any Spot instance requests in this region.

Click on the "Request Spot Instances" button to request your first Spot

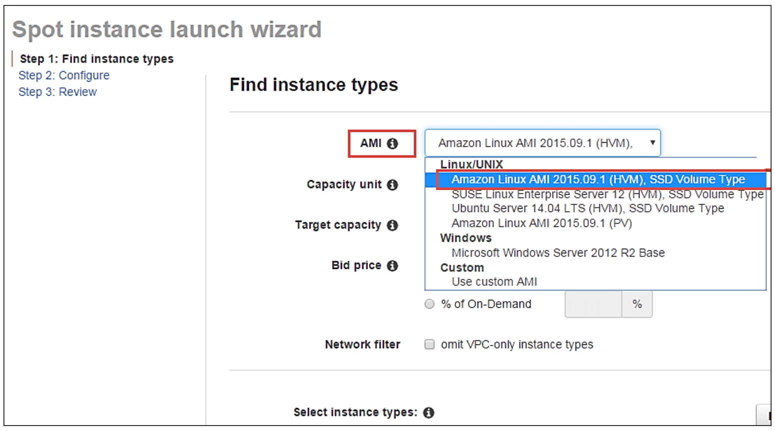
Spot instance launch wizard will open up. You can now go ahead with selecting the parameters and the instance configuration.

Find Instance Types

The first step for spot instance is to "Find instance types."

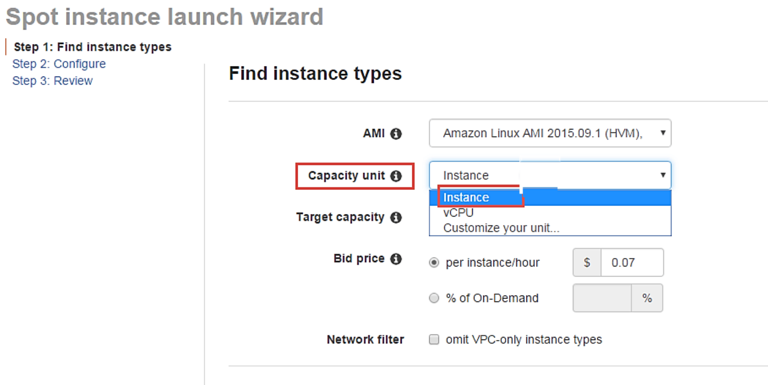


Step 1) Select an AMI- an AMI is a template consisting of the OS platform and software to be installed in the instance. Select your desired AMI from the existing list. We are selecting Amazon Linux AMI for this tutorial.


Spot instance launch wizard
Step 1: Find instance types
Step 2: Configure
Step 3: Review
Find instance types
AMI
Capacity unit
Target capacity
Bid price
Network filter
Select instance types:
Linux/UNIX
Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type
SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type
Ubuntu Server 14.04 LTS (HVM), SSD Volume Type
Amazon Linux AMI 2015.09.1 (PV)
Windows
Microsoft Windows Server 2012 R2 Base
Custom
Use custom AMI
○ % of On-Demand
%
□ omit VPC-only instance types

AMI: Amazon Linux AMI 2015.09.1 (HVM),

Step 2) Capacity Unit- a Capacity Unit is your application requirement. You may decide to launch an instance based on the instance type, vCPU or custom configuration like your choice of vCPU/memory/storage requirements. Here we are selecting an Instance.



AMI: Amazon Linux AMI 2015.09.1 (HVM),

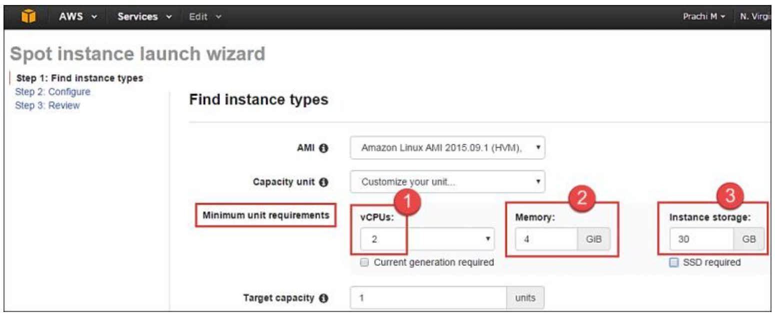
Capacity unit: Instance

Bid price: ● per instance/hour $ 0.07

○ % of On-Demand %

If you wish to customize the capacity, you can add your choice of

1. vCPU,
2. Memory and
3. Instance storage as below.



AMI: Amazon Linux AMI 2015.09.1 (HVM),

Capacity unit: Customize your unit...

Minimum unit requirements

vCPUs: 2

□ Current generation required

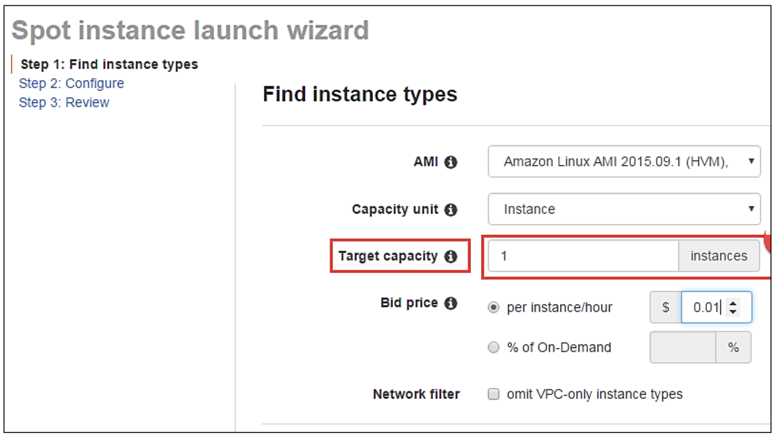
Memory: 4 GiB

Instance storage: 30 GB

□ SSD required

Target capacity: 1 units

Step 3) Target Capacity depicts how many spot instances you wish to maintain in your request. Here we are selecting one.



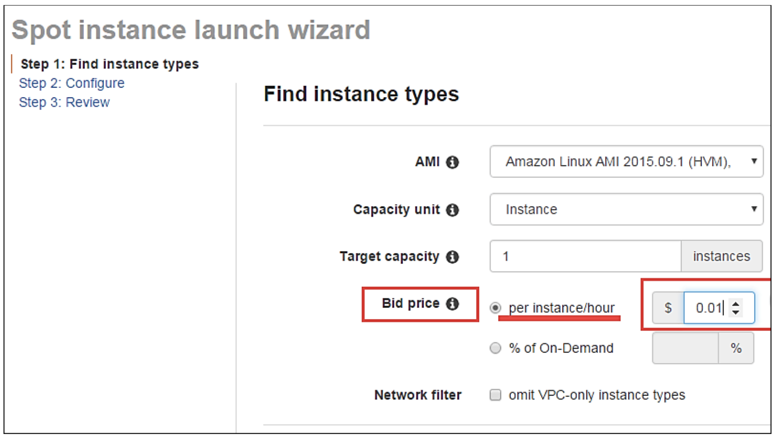
Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Find instance types

AMI: Amazon Linux AMI 2015.09.1 (HVM),
Capacity unit: Instance
Target capacity: 1 instances
Bid price: ● per instance/hour $ 0.01
○ % of On-Demand %
Network filter: □ omit VPC-only instance types

Step 4) Bid Price – this is the maximum price we are ready to pay for the instance. We are going to set a particular price per instance/hour. This is the simplest to calculate based on our business requirement. We will see ahead how we should determine the bid price so that our bid price always remains high and doesn't exceed the spot price so that our instance keeps running.



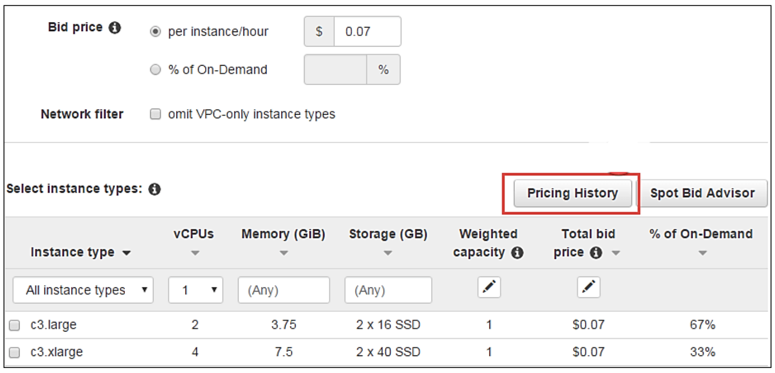
Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Find instance types

AMI: Amazon Linux AMI 2015.09.1 (HVM),
Capacity unit: Instance
Target capacity: 1 instances
Bid price: ● per instance/hour $ 0.01
○ % of On-Demand %
Network filter: □ omit VPC-only instance types

just below the bid price you can see a button of Pricing History. Click on that as shown below.



Bid price: ● per instance/hour $ 0.07
○ % of On-Demand %
Network filter: □ omit VPC-only instance types

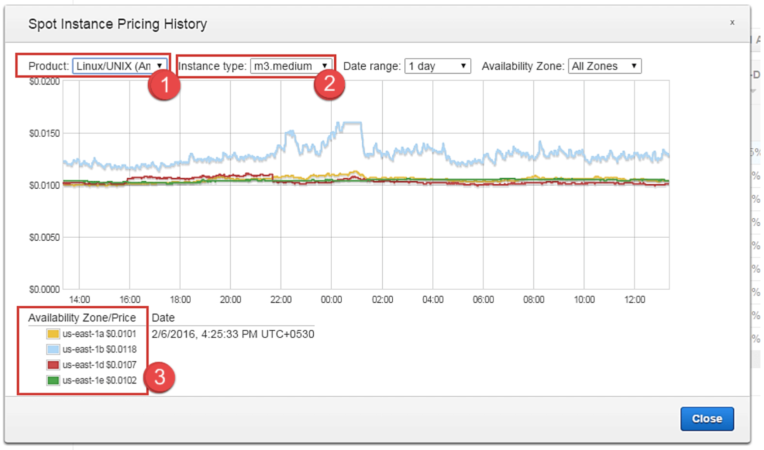
Select instance types:

	Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity	Total bid price	% of On-Demand
	All instance types	1	(Any)	(Any)			
□	c3.large	2	3.75	2 x 16 SSD	1	$0.07	67%
□	c3.xlarge	4	7.5	2 x 40 SSD	1	$0.07	33%

Here in Pricing History, we can see a graph depicting instance pricing trends with historical data. You can select the parameters and get an idea of the pricing of our desired instance over a period of time.

1. Select the product. We have selected our Linux AMI.
2. Select the instance type. We have selected m3.medium.
3. Note the average prices for over a day here.

Thus, from the chart below, we can see that the instance type that we are planning to provision lies in the pricing range of $0.01xx, and it seems that Availability Zone 'us-east 1a' has the lowest price.



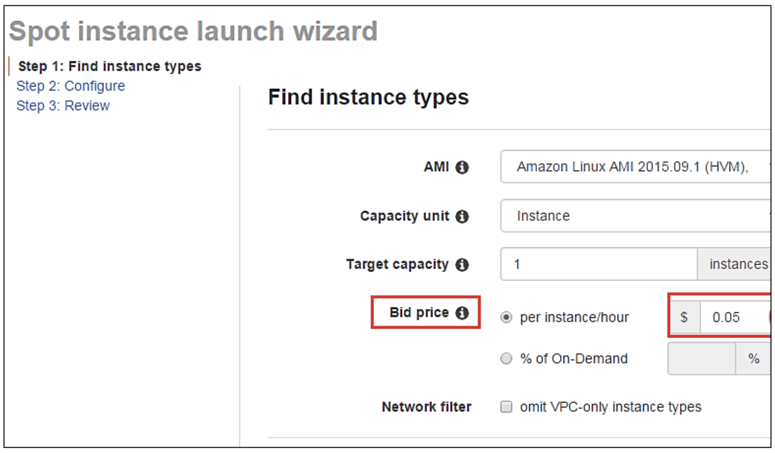
Product: Linux/UNIX (An ▾ Instance type: m3.medium ▾ Date range: 1 day ▾ Availability Zone: All Zones ▾

Availability Zone/Price	Date
🟨 us-east-1a $0.0101	2/6/2016, 4:25:33 PM UTC+0530
🟦 us-east-1b $0.0118	
🟥 us-east-1d $0.0107	
🟩 us-east-1e $0.0102	

cont. to step 4.

So let's come back to our step of quoting a bid price.

For the sake of maintaining our instance always available and if it falls within our budget, we can quote a higher bid price. Here we have quoted a slightly higher price of $0.05.



Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Find instance types

AMI: Amazon Linux AMI 2015.09.1 (HVM),

Capacity unit: Instance

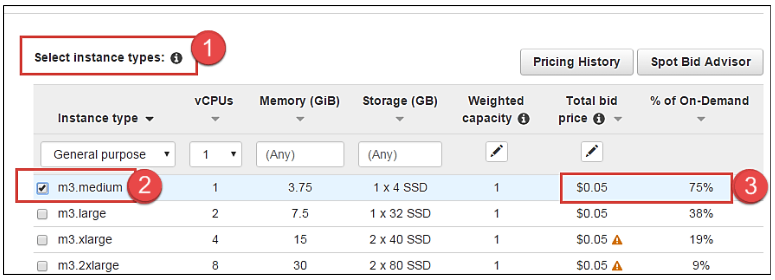
Target capacity: 1 instances

Bid price: ● per instance/hour $ 0.05
○ % of On-Demand %

Network filter: □ omit VPC-only instance types

You can see some trends in the wizard itself.

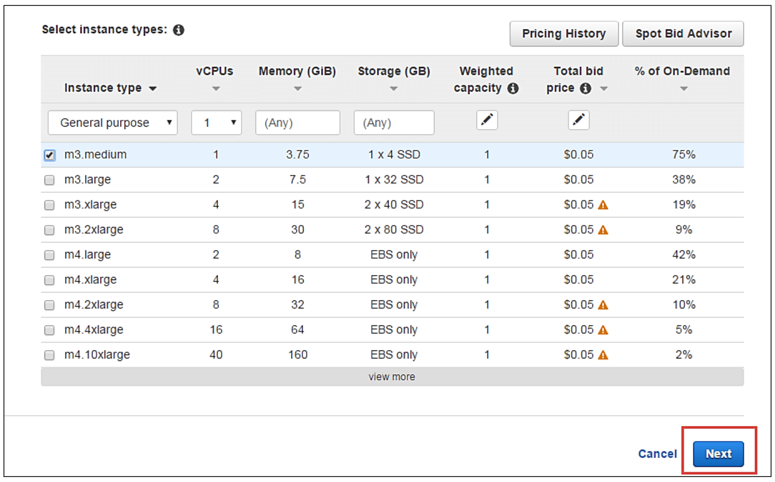
1. Note the instance types section
2. Select the instance type that we are planning to provision
3. Note the price that we are planning to bid. % of on-demand shows us that our quoted price is 75% of the on-demand price for the same instance type. This means we are saving 25% per hour as compared to an on-demand instance. You can further lower the price and save costs drastically.



Select instance types:

Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity	Total bid price	% of On-Demand
General purpose	1	(Any)	(Any)			
☑ m3.medium	1	3.75	1 x 4 SSD	1	$0.05	75%
□ m3.large	2	7.5	1 x 32 SSD	1	$0.05	38%
□ m3.xlarge	4	15	2 x 40 SSD	1	$0.05 ⚠	19%
□ m3.2xlarge	8	30	2 x 80 SSD	1	$0.05 ⚠	9%

Step 5) Once we are done looking at the trends and quoting our bid price, click on next.



Select instance types:

	Instance type	vCPUs	Memory (GiB)	Storage (GB)	Weighted capacity	Total bid price	% of On-Demand
	General purpose	1	(Any)	(Any)			
☑	m3.medium	1	3.75	1 x 4 SSD	1	$0.05	75%
□	m3.large	2	7.5	1 x 32 SSD	1	$0.05	38%
□	m3.xlarge	4	15	2 x 40 SSD	1	$0.05 ⚠	19%
□	m3.2xlarge	8	30	2 x 80 SSD	1	$0.05 ⚠	9%
□	m4.large	2	8	EBS only	1	$0.05	42%
□	m4.xlarge	4	16	EBS only	1	$0.05	21%
□	m4.2xlarge	8	32	EBS only	1	$0.05 ⚠	10%
□	m4.4xlarge	16	64	EBS only	1	$0.05 ⚠	5%
□	m4.10xlarge	40	160	EBS only	1	$0.05 ⚠	2%

Configure the Spot instance

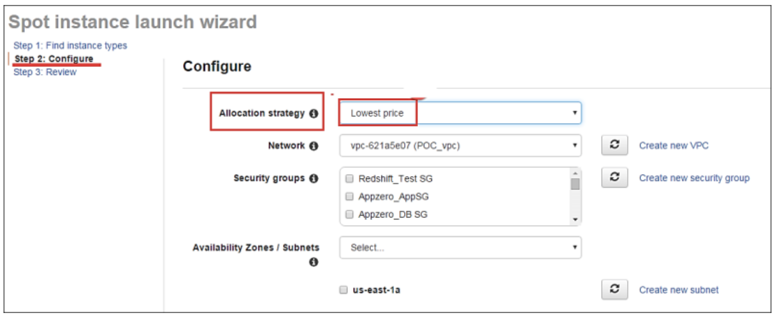
Our next step is to configure the instance, in this step of the wizard, we'll configure instance parameters like VPC, subnets, etc.

Let's take a look.

Step 1) Allocation Strategy – it determines how your spot request is fulfilled from the AWS's spot pools. There are two types of strategies:

- Diversified – here, spot instances are balanced across all the spot pools
- Lowest price – here, spot instances are launched from the pool which has lowest price offers

For this tutorial, we'll select Lowest Price as our allocation strategy.



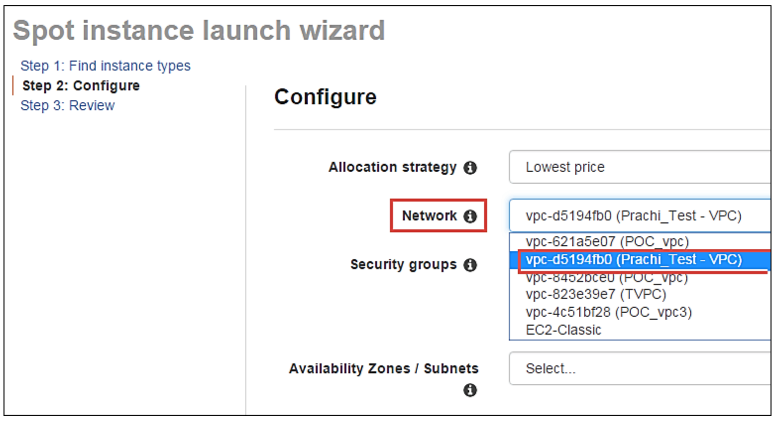
Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Configure

Allocation strategy: Lowest price
Network: vpc-621a5e07 (POC_vpc)
Security groups: □ Redshift_Test SG, □ Appzero_AppSG, □ Appzero_DB SG
Availability Zones / Subnets: Select...
□ us-east-1a

Step 2) Select the VPC- we'll select from the list of available VPCs that we have created earlier. We can also create a new VPC in this step.



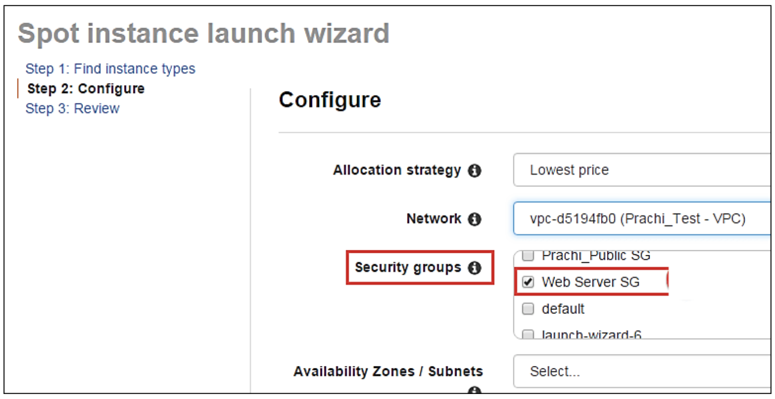
Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Configure

Allocation strategy: Lowest price
Network: vpc-d5194fb0 (Prachi_Test - VPC)
vpc-621a5e07 (POC_vpc)
vpc-d5194fb0 (Prachi_Test - VPC)
vpc-8452bce0 (POC_vpc)
vpc-823e39e7 (TVPC)
vpc-4c51bf28 (POC_vpc3)
EC2-Classic
Security groups
Availability Zones / Subnets: Select...

Step 3) Next we'll select the security group for the instance. We can select an already existing SG or create a new one.



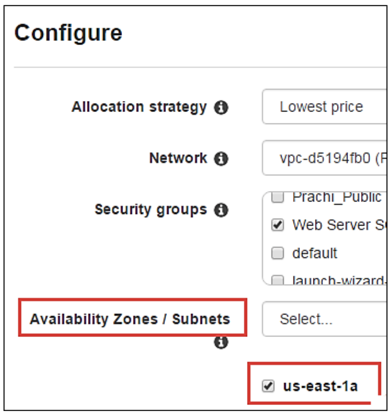
Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Configure

Allocation strategy: Lowest price
Network: vpc-d5194fb0 (Prachi_Test - VPC)
Security groups: □ Prachi_Public SG, ☑ Web Server SG, □ default, □ launch-wizard-6
Availability Zones / Subnets: Select...

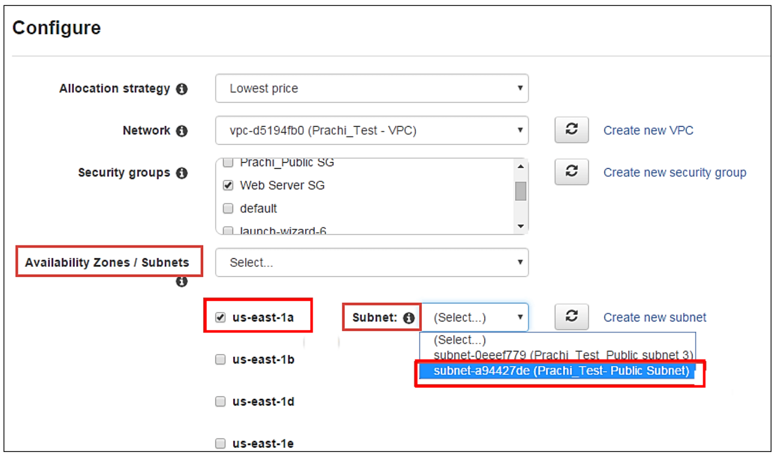
Step 4) Availability Zone- we'll select the AZ where we want to place our instance based on our application architecture. We are selecting AZ- us-east-1a.



Configure

Allocation strategy: Lowest price
Network: vpc-d5194fb0 (F
Security groups: □ Prachi_Public, ☑ Web Server S, □ default, □ launch-wizard
Availability Zones / Subnets: Select...
☑ us-east-1a

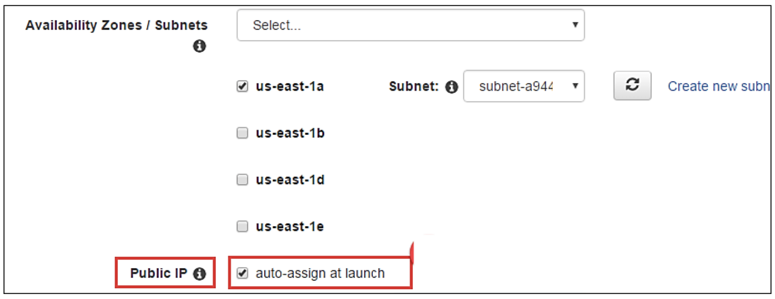
Step 5) Subnets- we are going to select the subnet from our list of already available list.



Configure

Allocation strategy: Lowest price
Network: vpc-d5194fb0 (Prachi_Test - VPC) — Create new VPC
Security groups: □ Prachi_Public SG, ☑ Web Server SG, □ default, □ launch-wizard-6 — Create new security group
Availability Zones / Subnets: Select...
☑ us-east-1a — Subnet: (Select...) — Create new subnet
(Select...)
subnet-0eeef779 (Prachi_Test_Public subnet 3)
subnet-a94427de (Prachi_Test- Public Subnet)
□ us-east-1b
□ us-east-1d
□ us-east-1e

Step 6) Public IP- we'll choose to assign the instance a public IP as soon as it launches. In this step, you can choose if you want AWS to assign it an IP automatically, or you want to do it manually later. You can enable/ disable 'Auto assign Public IP' feature here likewise.

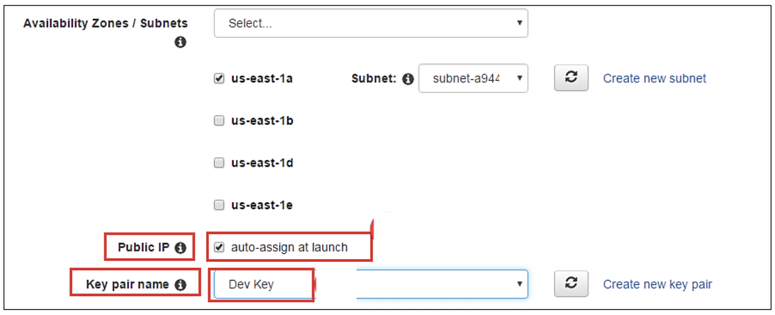


Availability Zones / Subnets: Select...
☑ us-east-1a — Subnet: subnet-a944 — Create new subn
□ us-east-1b
□ us-east-1d
□ us-east-1e
Public IP: ☑ auto-assign at launch

Step 7) Key pair- A key pair is a set of public-private keys.

AWS stores the private key in the instance, and you are asked to download the private key. Make sure you download the key and keep it safe and secured**; if it is lost you cannot download it again.**

After selecting public IP, here we are selecting a key which we already have created in our last tutorial.



Availability Zones / Subnets: Select...

☑ us-east-1a Subnet: subnet-a944 Create new subnet

☐ us-east-1b

☐ us-east-1d

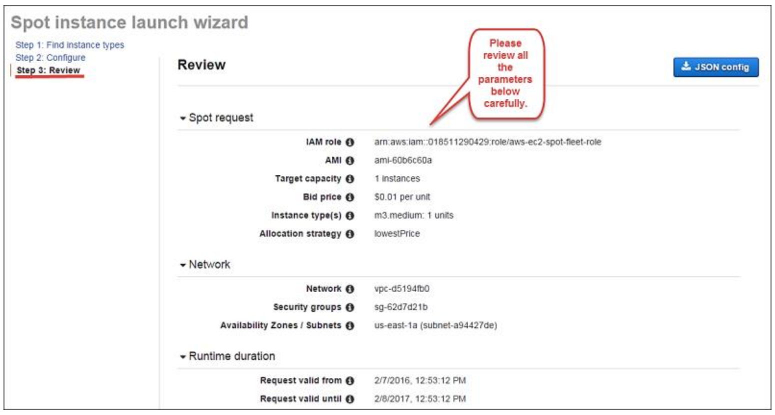
☐ us-east-1e

Public IP: ☑ auto-assign at launch

Key pair name: Dev Key Create new key pair

Review your Spot instance

Once we are done configuring our spot instance request in the 2 steps earlier in our wizard, we'll take a look at the overall configuration.



Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Review

Please review all the parameters below carefully.

Spot request

IAM role: arn:aws:iam::018511290429:role/aws-ec2-spot-fleet-role
AMI: ami-60b6c60a
Target capacity: 1 instances
Bid price: $0.01 per unit
Instance type(s): m3.medium: 1 units
Allocation strategy: lowestPrice

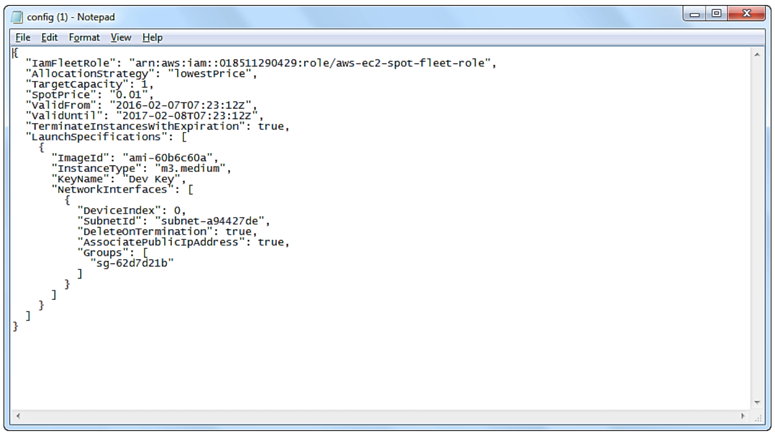
Network

Network: vpc-d5194fb0
Security groups: sg-62d7d21b
Availability Zones / Subnets: us-east-1a (subnet-a94427de)

Runtime duration

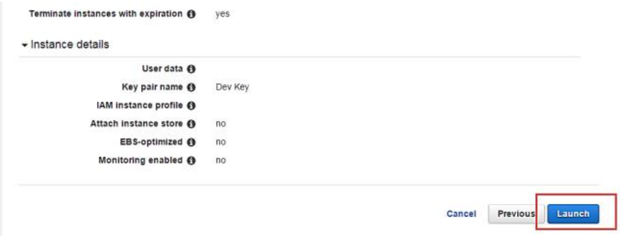
Request valid from: 2/7/2016, 12:53:12 PM
Request valid until: 2/8/2017, 12:53:12 PM

1. **We can also download a JSON file with all the configurations. Below is our JSON file.**



```
{
  "IamFleetRole": "arn:aws:iam::018511290429:role/aws-ec2-spot-fleet-role",
  "AllocationStrategy": "lowestPrice",
  "TargetCapacity": 1,
  "SpotPrice": "0.01",
  "ValidFrom": "2016-02-07T07:23:12Z",
  "ValidUntil": "2017-02-08T07:23:12Z",
  "TerminateInstancesWithExpiration": true,
  "LaunchSpecifications": [
    {
      "ImageId": "ami-60b6c60a",
      "InstanceType": "m3.medium",
      "KeyName": "Dev Key",
      "NetworkInterfaces": [
        {
          "DeviceIndex": 0,
          "SubnetId": "subnet-a94427de",
          "DeleteOnTermination": true,
          "AssociatePublicIpAddress": true,
          "Groups": [
            "sg-62d7d21b"
          ]
        }
      ]
    }
  ]
}
```

After we are done reviewing, we can proceed with the launching by clicking the Launch button as shown below.

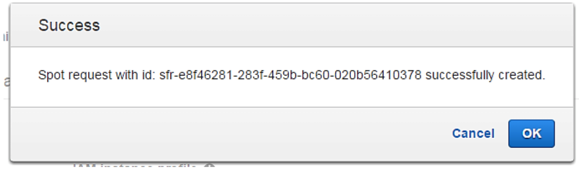


Terminate instances with expiration: yes

Instance details

User data	
Key pair name	Dev Key
IAM instance profile	
Attach instance store	no
EBS-optimized	no
Monitoring enabled	no

Once we select Launch, we can see a notification about the request getting created.

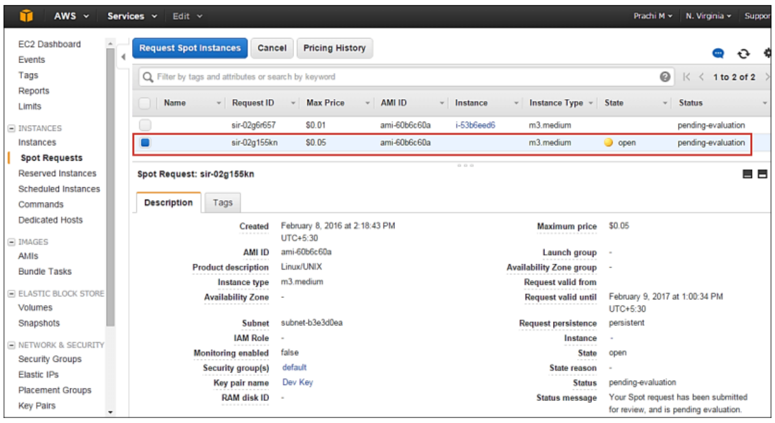


Success

Spot request with id: sfr-e8f46281-283f-459b-bc60-020b56410378 successfully created.

The spot request creation wizard will close, and the page will automatically direct back to the EC2 Dashboard.

You can see as shown below that the State of our request is 'open' which means that it is getting evaluated from the AWS's side. AWS EC2 will check if the required instance is available in its spot pool.



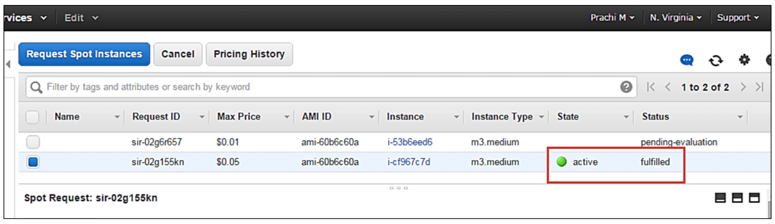
Name	Request ID	Max Price	AMI ID	Instance	Instance Type	State	Status
	sir-02g6r657	$0.01	ami-60b6c60a	i-53b6eed6	m3.medium		pending-evaluation
	sir-02g155kn	$0.05	ami-60b6c60a		m3.medium	open	pending-evaluation

Spot Request: sir-02g155kn

Description | Tags

Created	February 8, 2016 at 2:18:43 PM UTC+5:30	Maximum price	$0.05
AMI ID	ami-60b6c60a	Launch group	-
Product description	Linux/UNIX	Availability Zone group	-
Instance type	m3.medium	Request valid from	
Availability Zone	-	Request valid until	February 9, 2017 at 1:00:34 PM UTC+5:30
Subnet	subnet-b3e3d0ea	Request persistence	persistent
IAM Role	-	Instance	-
Monitoring enabled	false	State	open
Security group(s)	default	State reason	-
Key pair name	Dev Key	Status	pending-evaluation
RAM disk ID	-	Status message	Your Spot request has been submitted for review, and is pending evaluation.

After a couple of minutes, you can see that the state is changed to 'active', and now our spot request is successfully fulfilled. You can note the configuration parameters below.



Name	Request ID	Max Price	AMI ID	Instance	Instance Type	State	Status
	sir-02g6r657	$0.01	ami-60b6c60a	i-53b6eed6	m3.medium		pending-evaluation
	sir-02g155kn	$0.05	ami-60b6c60a	i-cf967c7d	m3.medium	active	fulfilled

Spot Request: sir-02g155kn

Synthesis

The issue of hosting infrastructure is an important one in the digital age because data and online services are now essential to enterprises. Companies frequently struggle with choosing between in-house servers and cloud hosting as they work to offer seamless client experiences while maintaining data security and scalability. In-house servers have long been the traditional approach, with companies investing in their own hardware, software, and dedicated IT teams. On the other hand, cloud hosting offers a flexible and scalable solution where computing resources are delivered as a service over the internet, eliminating the need for physical infrastructure on-site. Both options In-house Server vs. Cloud Hosting have their merits and drawbacks, making the decision a complex one. There are advantages cloud computing/hosting; (1) trade fixed expense for variable expense – Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can pay only when you consume computing resources, and pay only for how much you consume; (2) benefit from massive economies of scale – By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers is aggregated in the cloud, providers such as AWS can achieve higher economies of scale, which translates into lower pay as-you-go prices; (3) stop guessing capacity – Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often end up either sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little capacity as you need, and scale up and down as required with only a few minutes' notice; (4) increase speed and agility – In a cloud computing environment, new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, since the cost and time it takes to experiment and develop is significantly lower; (5) stop spending money running and maintaining data centers – Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking, and powering servers; (6) go global in minutes – Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide lower latency and a better experience for your customers at minimal cost.

In simple terms, EC2 lets you rent a virtual server or computer where you can host your application or data. There are benefits you can acquire in AWS;

- ***Easy to use*** - AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS's application hosting platform.
- ***Flexible -*** AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.
- ***Cost-Effective -*** You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more

information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

- ***Reliable -*** With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion-dollar online business that has been honed for over a decade.
- ***Scalable and high-performance -*** Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.
- ***Secure -*** AWS utilizes an end-to-end approach to secure and harden our infrastructure, including physical, operational, and software measures. For more information.

Reference

1. https://www.guru99.com/creating-amazon-ec2-instance.html
2. https://www.techtarget.com/searchcloudcomputing/tutorial/How-to-create-an-EC2-instance-from-AWS-Console
3. https://s3-us-west-2.amazonaws.com/uw-s3-cdn/wp-content/uploads/sites/149/2018/12/28193450/Curtis-Bray_Amazon_Introduction-to-Amazon-EC2.pdf
4. https://cloudkatha.com/how-to-create-ec2-instance-in-aws-step-by-step/
5. https://douran.academy/wp-content/uploads/ebooks/aws-for-beginners.pdf